MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 30, 1996

3. REPORT TYPE AND DATES COVERED
Final Rpt., 27 Sept. 93 to 30 Jun. 96

4. TITLE AND SUBTITLE
MANUFACTURING APPLIED RESEARCH AND DEVELOPMENT PROGRAM

5. FUNDING NUMBERS
DAAH04-93-G-0508

6. AUTHOR(S)
James Bentley, Hugh Chappelle, Richard Colbaugh, Leon Cox, Joseph Genin, Kristin Glass, Brian Lambert, Thomas Shay.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
New Mexico State University
P.O. Box 30001, Dept. 3450
Las Cruces, New Mexico 88003

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 32515.2-MS

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Five separate, but potentially related, areas of manufacturing research have been completed. These are: (1) Intelligent Automated Numerical Control Programming Systems; (2) Adaptive Control of Manufacturing Processes; (3) Contact Stress and Related Thermal/Mechanical Modeling; (4) Material-Cutting Tool Interfaces and Ion Implantation; and (5) Laser Instrumentation for Monitoring, Diagnosis, and Control. A record for laser measurement, using sinusoidal phase modulation, was achieved during task (5).

14. SUBJECT TERMS
Programming intelligence, turning process, automation, robotics, neural networks, contact stress, thermal/mechanical modeling, material interface(s), ion implantation, laser metrology, laser control.

15. NUMBER OF PAGES
341

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

MANUFACTURING APPLIED RESEARCH AND DEVELOPMENT PROGRAM

FINAL REPORT

HUGH L. CHAPPELLE II

JUNE 30, 1996

U.S. ARMY RESEARCH OFFICE

GRANT NO. DAAHO4-93-G-0508

NEW MEXICO STATE UNIVERSITY

APPROVED FOR PUBLIC RELEASE;

DISTRIBUTION UNLIMITED

TABLE OF CONTENTS

INTELLIGENT AUTOMATED NUMERICAL CONTROL PROGRAMMING

Leon Cox

Department of Mechanical Engineering
New Mexico State University

Final Report for Task 3.1 of Applied Manufacturing Research
Submitted to the Army Research Office

February 1996

Intelligent Automated Numerical Control Programming Systems Research Task 3.1 Final Report

EXECUTIVE SUMMARY

Effective storage, processing and exchange of product data is essential to improving manufacturing operations. National efforts to standardize the manner in which a product is described include efforts by the Computer Aided Logistics Support (CALS), the International Graphics Exchange Standard (IGES), and the Product Definition Exchange Specification/Standard for Exchange of Product Data (PDES/STEP). These efforts guided the manner in which the project progressed. Prof. Leon Cox participated in the '94 Air Force CALS Testbed (AFCTB) project utilizing IGES V5.1 translator to download product data for use with the developing Turning Assistant to develop the CNC program for part number 7307165 pin and shipped the finished pin to the CALS conference for the assembly demonstration with no noted problems. Dr. David Culler attended the CALS conference, Prof. Cox attended Autofact conference and discussed PDES/STEP efforts with Gary Patelski of General Motors (GM), who is heading efforts there.

The dissertation work by Dr. David Culler completes the objectives to develop an automated turning package capable of handling a variety of rotational features that can be extended to other machining and process planning activities. The work by Prof. Leon Cox and Dr. Amjed Al-Ghanim in artificial neural networks (ANN's) addresses the objective to develop a formalized knowledge base of "industrial strength" turning strategies. As the several papers point out are useful tools for collecting, storing and utilizing ANN's in an automated CNC programming environment. Further, the turning assistant was able to make calls to the ANN's and obtain parameters useful in the process planning of a product manufacture. This work can never completely reach completion to provide a complete knowledge base because there is a limitless supply of data from machinists; however, the format by which the ANN's have been developed provide the means for individual companies to develop a rich subset. And finally, work by Kathy Lueders, MSIE candidate, to develop links to commercial knowledge bases is progressing and an unpublished paper describing her work is included.

PAPERS INCLUDED

Feb. 94, SYSCAP Proceedings: Cox, Al-Ghanim, "A Neural Network Approach for Computer-Aided Process Planning of CNC Milling Parameters and Strategies ".

May 94, IERC Proceedings: Cox, Al-Ghanim, "Neural Networks Applied to Computer-Aided Process Planning".

May 94, Ph.D. Dissertation, Culler, David E., "The Turning Assistant: Automated Planning for Numerical Control Lathe Operations".

Feb. 95, Proceeding of the 1st World Congress on Intelligent Manufacturing Processes and Systems: Cox, Al-Ghanim, Culler, "Knowledge Acquisition Techniques for Intelligent Machining Assistants".

Mar. 95, Proceedings of the XVII Conference on Computers and Industrial Engineering: Cox, Al-Ghanim, Culler, "A Neural Network-Based Methodology for Machining Knowledge Acquisition". Also appears in Pergamon Press Computers industrial engineering, Vol. 29 pp 217-220, 1995.

Nov. 95, Intelligent Engineering Systems Through Artificial Neural Networks, Cox, Al-Ghanim, Culler, "Machining Strategies Collected, Stored and Utilized with Artificial Neural Networks", ISBN 0-7918-0048-2, ASME Press, 1995.

Dec. 95, Cox, Al-Ghanim, "A Process Planning Technique Utilizing Neural Networks to Determine CNC Milling, Drilling and Turning Parameters and Strategies", (Abstract, Not Published).

Dec. 95, Lueders, Kathryn, "Utilization of Knowledge Base, Process Planning Tool and Neural Net Inputs to a Machining System", (Course paper and basis for Thesis).

TASK 3.1

PUBLICATIONS AND TECHNICAL REPORTS PUBLISHED

A Neural Network Approach for Computer-Aided Process Planning of CNC Milling Parameters and Strategies

Amjed Al-Ghanim, PhD Candidate and Leon Cox, Assoc. Prof.
New Mexico State University
Department of Industrial Engineering

ABSTRACT

Artificial neural systems are becoming an integral part of the computer integrated manufacturing environments, especially in a market place where improvements must take place in the form of increased productivity. The function of process planning is a prominent one that has direct impact on overall manufacturing productivity. This paper is an initial investigation into the potential of applying artificial neural paradigm to the selection of machining parameters of a milling process. The viability of this approach stems from the ability of neural nets to handle ill-structured problems, and their capability of generalization. Our simulation results, applied to tool material selection, show a high potential for the development of neural network modules for similar process planning functions.

1.0 INTRODUCTION

Process Planning (PP) can be defined as the subsystem responsible for converting design data into work instructions. It represents the link between engineering design and shop floor manufacturing, and determines the manufacturing operations required to transform a part from a rough state to a finished state specified by the engineering drawing. More specifically, it is a function which maps the design features to manufacturing features [1]. As related to machining, process planning consists of a series of tasks interpreting the product model including selection of machine tools, tool sets, setups (e.g. design of fixtures), and machining sequences (e.g. generation of NC programs) [2]. As such, process planning is an involved activity that has a large number of input variables describing the part to be produced and the production resources.

As established, there are two systematic computer-aided process planning strategies; namely the *variant planning* and the *generative planning*. The distinguish ing feature of the variant planning strategy is that former plans are retrieved and modified for new parts, while generative planning is a strategy that strives to create a new plan for a part from scratch based on analyzing part geometry and other related specifications [1].

Artificial Intelligence (AI) technology, Group Technology (GT), [3], and Expert Systems (ES) technology [4,5,6] have been employed to assist process planning. Rule-based expert systems based on experience extracted from machinists have been built to generate process plans. Various methodologies have been employed to implement generative CAPP expert systems including among others, case-based reasoning [2], feature-based recognition [7,8], and solid modeling techniques [9,10].

If a product model existed that contained all processing parameters necessary for the manufacture of the product then a subset of these would contain information regarding any machining that would need to take place during a particular interval of development. Virtually all products require a chip producing machine tool in some stage of manufacture. Today much of the chip producing is being accomplished through computer numerical control (CNC).

In most cases during the planning for the CNC machining operations, it is necessary to consult an expert. Whether that expert is a machinist, part programmer or a knowledge base, the specific set-up and run parameters for the machining process must be explicitly determined.

We are focusing on the ability of the human (and in the long run the ability of a pattern recognition system) to identify a feature that needs to be machined such as a pocket, notch, slot, etc. Once identified that feature will have parameters and strategies developed specific to its nature. Thereby, taking the larger problem and simplifying it.

Most research efforts as described have focused on utilizing expert systems technology to implement process planning systems. These systems offer assistance based on capabilities ranging from fast retrieval of existing plans to generating plans based on interpreting product model geometry. Actually, expert systems have shown success in automating certain aspects of process planning. However, with the increasing level of complexity of manufacturing environments, and the increasing demand for achieving levels of integration, expert systems methodology suffers some limitations. First, when domain knowledge is complex and intuitive as in the case of process planning, the knowledge acquisition phase of system development becomes a real bottleneck because it is not always possible to extract and represent knowledge in an explicit form [2,6,11]. Second, when the system rule base gets larger, the knowledge validation process becomes difficult and prone to errors [12]. Finally, a dynamic manufacturing environment requires utilization of fast adaptive systems. However, expert systems, due to explicit knowledge representation, show limited adaptive capabilities, and they do not tolerate missing or inaccurate data.

To compensate for some of these limitations and enhance the expert systems approach, this paper describes a methodology of process planning for CNC milling based on the concept of artificial neural networks. First, process planning is cast as a pattern recognition problem, and viewed as a mapping function between design features and machining features. Second, a neural network model is developed for automating certain aspects of process planning decisions such as cutting tool material, tool entry method, and cutting strategies. Simulation results are shown for appropriate network parameters.

2.0 ARTIFICIAL NEURAL NETWORKS

There has been an increase of interest in 'brain style computing' in terms of artificial neural networks (ANNs). ANNs are parallel distributed processing architectures composed of a huge number of small interacting elements that are massively interconnected. Each of these elements sends excitatory and inhibitory signals to other units that in turn update their behaviors based on these received messages. ANNs emulate the functionality of the human brain that implicitly stores knowledge in the interconnection weights, and not in the neurons themselves [13]. Learning is thus achieved by modifying the connection weights between elements. Feedforward neural nets acquire knowledge through training, where by repeatedly presenting *sample* cases to the net, its interconnection weights change accordingly to model the representation of the cases [14]. Knowledge is stored in the final interconnections values. As such neural network methodology would ease the knowledge acquisition bottleneck that is hampering the creation of expert system [15]. The key to a neural network's power is the inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization. A typical network consists of a set of processing elements (PE) grouped hierarchically in layers and interconnected in some fashion. These PEs sum N weighted inputs and transfer that outcome through a mathematical function. Figure 1 shows a processing element and a three layer neural network. One of the most popular neural paradigms, and the one that has been used in this project, is the multilayer perceptron trained using the Error Back Propagation Training Algorithm (EBPTA) [16]. Within the manufacturing environment ANNs have been applied to quality control, process control, part-family classification [17,18,19,20,21] Neural network models have been also used for optimizing cutting parameters of turning operations [22].

Figure 1. Generic Neural Network Components

2.1 NEURAL NETWORK DESIGN CONSIDERATIONS

One of the advantages of the backpropagation algorithm is its applicability to a large number of network design architectures; that is, the choices for the number of layers, interconnections, learning constants, and data representation [16]. The problem of selecting network parameters is under intensive research and there are no conclusive answers available [15]. Analysis of this problem takes the form of experimenting with different design choices and performing simulations, the results of which give guidelines for final parameter selection. The design factors considered here are: learning rate, and number of neurons in a hidden layer. The number of layers in a network can only be determined through simulation. However, the literature suggests a maximum of three to four layers (i.e. one to two hidden layers); as more than three layers will improve performance very little at the expense of high convergence time [24]. The number of input neurons is simply identical to the dimension size of the input vector. In cases where the number of input neurons is less than input vector size, then the number of training samples has to be increased. The number of output neurons is usually selected to be the number of classes. In such a scheme, called local representation, the network performs as a decoder. The number of neurons in the hidden layer, *J*, can be determined by a simple (empirical) formula relating the number of separable decision regions, *M*, the size of the input vector *n*. [15]. However, in cases where the number of decision regions is not known (as in this case), the number of hidden neurons can initially be taken to be equal half the input vector dimension, *n*. To provide a satisfactory level of class separation each decision class should have enough number of examples (i.e. 5-10).

The effectiveness and convergence of the backpropagation algorithm depends significantly on the appropriate choice of the learning constant (η). In general, the optimum value of (η) depends on the problem being solved, and so can be determined experimentally. A true learning strategy calls for small values of (η), which will consequently increase convergence time. Larger values of (η) may cause the algorithm to overshoot and oscillate, thus preventing convergence. The momentum factor can be introduced to alleviate this problem, by allowing smaller values of (η) and a faster convergence process.

3.0 PROCESS PLANNING : A PATTERN RECOGNITION PROBLEM

Pattern Recognition (PR) can be characterized as an information mapping process taking place over a set of (metric/non-metric) spaces [14]. An abstract view of the PR description problem is shown below in Figure 2. A relationship G_i stands as a mapping between class-membership space, *C*, and a pattern space, *P*. Each Class w_i corresponds to a subset of patterns in the pattern space, where the i^{th} pattern is denoted as p_i. These pattern spaces may overlap allowing patterns from different classes to share the same attributes. Figure 2 shows also another mapping from the pattern space to an observation space, with feature patterns m_i, via the relationship M. All patterns pertaining to the different spaces are represented in an abstract form by vectors, and any given space may assume a specific dimension, n, and thus be represented generically as R^n.

A relationship can formally be defined as $M: R^n -\!\!> R^m$; that is a mapping from a space of dimension n to a space of dimension m. Using this concept, the characterization of many PR problems is simply that, given a observation vector m_i, a method is desired to identify and invert mapping M and G_i for all i. As seen from Figure 2, different pattern vectors may be associated with the same observation vector, and single class membership can point to more that one pattern. This characteristic suggests a decision problem with *ambiguity or vagueness*. Pattern recognition techniques; namely statistical, structural, and artificial neural are employed to overcome vagueness. Depending on the nature of the problem (the pattern generating environment) and on how well it can be structured, a suitable PR technique can be applied. For example, statistical PR can be used when patterns come from a statistically characterized environment. However, quantitative precise descriptions of some complex decision problems may be largely unknown and a detailed characterization may be hopeless. A viable alternative is to treat the problem from an input/output or 'black box' viewpoint [14]. A Neural pattern recognizer is such a good example of black box systems. Because process planning (PP) is such an ambiguous and ill-structured problem (at least in certain aspects), it seems reasonable to approach PP using neural network PR techniques.

Figure 2. PR: Mapping in an Abstract Form

Process planning can be cast as a pattern recognition problem; that is set of mapping from one space(s) to another space(s). First, as explained earlier, process planning for CNC machining is a multivariable input/output process, including among others part dimensions, material, surface finish, fixture rigidity, tool material, spindle speed, feed rate, depth of cut, etc.. Therefore, to facilitate analysis, one should precisely identify the output decision quantities and corresponding input decision quantities. In this case, as related to milling, the output decision variable is cutting tool material type. Other factors being considered are the strategy for entry of the tool into the material and the cutting strategy once the material is entered.

The input decision variables pertaining to the part to be manufactured are: feature types, perimeter, length, width, depth, corner radius, material, tolerance, and finish as shown in figure 3. Other input decision variables are machinist preference and delivery requirement of the part. Each of the output decision variables depends on a mixture of the input variables, and possibly on some other input variables. The mixture of the input variables forms the input space, or in pattern recognition terminology the observation space. The output variables forms the output space, or likewise the decision space. *The idea of the 'black box' is to collapse the effects of individual input variables into a combined input effect, with no major emphasis on any particular input variable.* The concept permits capturing inexact relationships between individual inputs and outputs. Figure 4 is a schematic view of a 'black box' of a sub-process planner. Here, the output decision variable is the cutting tool material.

Figure 3. A Pocket Feature

Figure 4. A 'Black Box' View of the Tool Material Type Selector

4.0 A NEURAL NETWORK PROCESS PLANNER

The research methodology adopted aimed at exploring the utilization of neural network models as supportive decision-making tools in machining process planning. To carry out this objective, process parameters have been identified, as related to milling processes. Table 1 shows the input and output parameters for three network models.

Table 1. Neural Network Machining Strategy Models

Network 1.	Output decision variable: Tool material type	
	Values (classes) of output:	High speed steel Carbide Ceramic
	Input variables:	Feature type Depth of cut Part material hardness Machinability Fixture rigidity M/C max rpm, feed, horsepower
Network 2.	Output decision variable: Entry strategy	
	Values (classes) of output:	Free entry Drill hole Ramp Plunge Helix
	Input variables:	Feature type, # of Islands, smallest gap Feature Width, Depth, Length, Perimeter Part material hardness Fixture rigidity M/C max rpm, feed, horsepower Machinability Surrounding geometry Geometry near entry.
Network 3.	Output decision variable: Cutting strategy	
	Values (classes) of output:	Collapse-in Collapse-out Constant angle Finish Rough Finish and rough
	Input variables:	Feature type, # of Islands, smallest gap Feature Length, Width, Depth, Perimeter Corner rad(convex) Corner rad(concave) Part matrl hardness Side tolerance Surface finish rms Fixture rigidity M/C max rpm, feed, horsepower Machinability Surrounding geometry Geometry near entry Base tolerance

It should be noted that the output decision variables were chosen such that their values can be grouped into a limited number of classes. Actually, this the basic idea of utilizing neural nets as pattern recognizers. During the learning phase, the network learns all possible classes and corresponding regions, and consequently, during the testing phase, the network recognizes a new input case (vector) by assigning it to the closest decision class. A new case represented to the network as a vector may be corrupted by noise (e.g. machinist preference); however, the network will recommend the closest possible solution in terms of the output variable.

The development process of the network consists of two main steps. First, a preliminary data set was obtained, and an initial network architecture was selected for training to make sure that a neural model of the problem can be generated. Second, a sensitivity analysis procedure was conducted to determine optimal network design parameters. The data collection phase was carried out in Integrated Manufacturing Systems Laboratory at New Mexico State University from interfiews with machinists. Forty examples were constructed for each decision output variable (i.e. about 10-15 examples for every output class) which provided a good level of discrimination between classes.

As shown in Table 1, input variables range in type and form. Some variables can intrinsically have numeric values (e.g. depth, hardness), while others (e.g. machinability and fixture rigidity) can best be described in linguistic form, such as high, medium or low. Since neural nets only operate on numeric data, such variables have been quantified so that they have numeric values between 0 and 1. The values of variables can range from .0001's (for feed) to 10k's (for machine rpm). During training of a network, the wide range can introduce a large amount of bias in the values of connection weights. To alleviate this problem, all input variables have normalized so that their values fall in the range 0 and 1. This range is actually a preferable choice for network operation; as the activation function (unipolar sigmoid) assumes the same range, thus resulting in moderate values of the weights. The cuttoff value for classification is (.80).

5.0 NETWORK TRAINING AND RESULTS

For this paper only the results of the tool material selection strategy are presented since it is a primary decision. Training was carried out using a two-layer network with a momentum factor of (.9), a learning factor of (.1), and target error of (5%). Several runs were conducted for various numbers of hidden neurons as shown in Table 2, for the tool material network. Guided by these results the number of hidden neurons was selected to be 16.

Table 2 # of neuron vs. convergence cycles (η =.1)

network #	# of hidden neurons	# of converg. cycles
1.	5	1442
2.	8	2161
3.	11	1461
4.	13	1156
5.	16	817

Then using this number of hidden neurons, the network was retrained with different values of the learning constant, as shown in Table 3.

Table 3 Learning constant vs. convergence cycles (hidden neurons=15)

learning constant(η)	convergence cycles
0.10	817
0.20	631
0.30	501

0.35	593
0.40	1034

One of the important performance measures of network operation is the percentage of classification errors. The networks designed were tested first, using the training examples, and second, using a new set of 20 known cases. The performance of the network exceeded the expectations; as the network was unable to recognize at most two of the forty training examples, for different learning values. When used in the testing mode, the network performance ranges from a low 69% for a learning constant of .30 to a high 92% for a learning constant of 0.10 correct classifications of the exposed cases. These results are shown in Table 4. These results are obtained for a target error value of (5%).

Table 4 Network classification performance

	hidden nodes	learning constant(η)	correct classification (%)
(Training examples)	5	.10	95%
	11	.10	95%
	16	.10	100%
	5	.20	90%
	11	.20	95%
	16	.20	97%
	5	.30	95%
	11	.30	95%
	16	.30	95%
(Testing examples)	5	.10	88%
	11	.10	90%
	16	.10	92%
	5	.20	80%
	11	.20	85%
	16	.20	88%
	5	.30	80%
	11	.30	76%
	16	.30	69%

6.0 CONCLUSION

We have shown the viability of utilizing the Neural Networks approach for assisting the process planner or CNC programmer in obtaining machining set-ups and strategies. We were able to handle the uncertainty and vagueness that is inherent in the minds of the "experts". It also shows strong promise as a method of storing knowledge and being retrievable in a rapid manner. As new examples become proven methodology for an industry, the Neural Net is easily retrained to handle updated information. The optimal design of a Neural Network architecture is achieved through running simulations and in our case the "best" was sixteen hidden nodes and 0.1 learning factor. Future work is planned in fuzzifying the linguistic type inputs and verifying the strategic outputs on CNC machine tools.

7.0 ACKNOWLEDGEMENTS

The research findings presented in ths paper were were supported in part through contracts with Sandia National Laboratories Technology Transfer and the Army Research Office.

8.0 BIBLIOGRAPHY

1. D. Bedworth, R. Henderon, M. Wolfe, *'computer-Integrated Design and Manufacturing'*, McGraw-Hill, 1991.

2. L. Cox, D. Culler, and A. Al-Ghanim, '*Artificial Intelligence Application Utilization in Computer Numerical Control Manufacturing*', in Proc. of IIE Conference, Los Angeles, May 26,

1993, pp(490-496).

3. Kusiak A., '*Group Technology for Flexible Manufacturing Systems*', Handbook of Flexible Manufacturing Systems, Academic Press, San Diego, CA, 1991.

4. T. Gupta, '*A State of the Art Survey of Expert Systems in Process Planning*', IIE Integrated System Conference in St. Louis, pp(269-274), Oct.30, 1988.

5. T. Gupta, '*A Survey of Expert Systems in Manufacturing and Process Planning: Current Development and it Future*', Computers and Industrial Eng., Vol.18, No.1, pp(69-80), 1990.

6. L. Alting, H. Zhang, '*Computer-Aided Process Planning: state-of-the-art survey*', International Journal of Production Research, Vol.27, No.1-4, pp(553-585), 1989.

7. Tsang J.P, and Brissuad D., '*A Feature-Based Approach to Process Planning*', Computers in Industrial Engineering, 1989.

8. van Houten, van't Erve, and Kals, '*A feature Based Computer Aided Process Planning*', Laboratorium voor Produktietechniek, Universiteit Twente, Netherlands, February 1989; in Proc. of the 21st CIRP International Seminar on Manufacturing Systems, Stockholm, 1989.

9. Y. Descote, J.C. Latombe, '*GARI: An Expert System for Process Planning, Solid Modeling Computers*', Plenum Press, New York, 1983.

10. G. Jared, '*Shape Features in Geometric Modeling, Solid Modeling by Computers*', Plenum Press, New York, 1983.

11. J. Giarratano, G. Riley, '*Experts Systems: principle and programming*', PWS publishing company, Boston, 1989.

12. Amjed Al-Ghanim, and Bassam Talhouni, '*A Diagnostic Expert System for IBM PCs and Compatibles*', Muta Journal for Scientific Research, Muata University, Jordan, (Accepted for publication).

13. Rumelhart D.E., McClelland J.L., and the PDP Research Group, '*Parallel Distributed Processing: Explorations in the Microstructure Cognition*', Volume 1, MIT Press, 1986.

14. Shalkof R.J., '*Pattern Recognition: Statistical, Structural, and Neural Approaches*', John Wiley & Sons, Inc., USA, 1992.

15. Zurada J.M., '*Introduction to Artificial Neural Systems*', West Publishing Company, USA, 1992.

16. Dayhoff J.E., '*Neural Networks Architectures: An Introduction*', van Nostrand International Company, New York, 1990.

17. S. Martinez, A. Smith, B. Bidanda '*A Neural Predictive Quality Model for Slip Casting Using Categorical Metrics*', in Proc. of IIE Conf., Los Angeles, May 26, 1993, pp(265-269).

18. Tomas Telsaco, '*Interfacing Process Quality Control and CIM Through Application of Artificial Neural Networks*', in Proc. of IIE Conference, Los Angeles, May 26, 1993, pp(280-284).

19. Godwin J. Udo, '*Neural Networks Applications in Manufacturing Processes*', Computers and Industrial Eng. Vol.23, No.1-4,pp(97-100), 1992.

20. Deborah F. Cook, Robert F. Shannon, '*A predictive Neural Network Modeling System for Manufacturing Process Parameters*', International Journal of Production Research, Vol.30, No.7, pp(1537-1550), 1992.

21. Kanda Charraprty, Utpal Roy, '*Connectionist Model for Part Family Classification*', Computers and Industrial Eng., Vol.24, No.2, pp(189-198), 1993.

22. Jun Wang, '*A Neural Network Approach to Multiple-Objective Cutting Parameter Optimization Based on Fuzzy Preference Information*', Computers and Industrial Engineering, Nos 1-4, pp(389-392), 1993.

23. Stephen Gallant, '*Neural Network Learning and Expert Systems*', MIT press, 1993.

Neural Networks Applied to Computer-Aided Process Planning

Amjed Al-Ghanim and Leon Cox
New Mexico State University
email: lcox@nmsu.edu

ABSTRACT

The function of process planning is a prominent one that has direct impact on overall manufacturing productivity. This paper presents results from preliminary investigations applying artificial neural networks ability to select machining parameters of a milling process and as a storage medium for the knowledge. The viability of this approach stems from the ability of neural nets to handle ill-structured problems, and their capability of generalization. Simulation results, applied to tool material selection and tool entry strategy, show a high potential for the development of neural network modules for similar process planning functions.

1.0 INTRODUCTION

Process planning is a function which maps the design features to manufacturing features [1]. It represents the link between engineering design and shop floor manufacturing, and determines the manufacturing operations required to transform a part from a rough state to a finished state specified by the engineering drawing. As related to machining, process planning consists of a series of tasks to interpret the product model including selection of machine tools, tool sets, setups (e.g. design of fixtures), and machining sequences (e.g. generation of NC programs) [2]. As such, process planning is an involved activity that has a large number of input variables describing the part to be produced and the production resources.

There are two systematic computer-aided process planning strategies; namely *variant planning* and *generative planning*. The distinguishing feature of the variant planning strategy is that former plans are retrieved and modified for new parts, while generative planning is a strategy that strives to create a new plan for a part, from scratch, based on analyzing part geometry and other related specifications [1]. Artificial Intelligence (AI) technology, Group Technology (GT), [3], and Expert Systems (ES) technology [4,5,6] have been employed to assist process planning. Rule-based expert systems based on experience extracted from machinists have been built to generate process plans. Various methodologies have been employed to implement generative CAPP expert systems including among others, case-based reasoning [2], feature-based recognition [7,8], and solid modeling techniques [9,10]. Expert systems have been shown to have specific limitations in these areas. [11,12]

Virtually all products require a chip producing machine tool in some stage of manufacture. Today much of the chip producing is being accomplished through computer numerical control (CNC). In most cases during the planning and programming stages for the CNC machining operations, it is necessary to consult an expert. If a product model that contains all processing parameters necessary for the manufacture of the product exists, then a subset of these would contain information regarding any machining that would need to take place during a particular interval of development. The specific set-up and run parameters for the machining process must be explicitly determined. Process planning for CNC machining is a multivariable input/output process considering, among others, part dimensions, material, surface finish, fixture rigidity, tool material, spindle speed, feed rate, depth of cut, etc.. As related to milling, the output decision variables can be, for instance, cutting *tool material type*, the *strategy for entry* of the tool into part material, and the *cutting strategy* once the part material is entered. For this investigation the first two decision variables have been investigated. We are focusing on two aspects, first, the ability of the human (and in the long run the ability of a pattern recognition system) to identify manufacturing features that need to be machined such as a pocket, notch, slot, etc. and second, the ability to store machining process planning information in a convenient manner for retreival. Once a feature is identified it will have parameters and strategies developed specific to its nature.

2.0 ARTIFICIAL NEURAL NETWORKS

There has been an increase in interest in 'brain style computing' in terms of artificial neural networks (ANNs). ANNs are parallel distributed processing architectures composed of a large number of small interacting elements that are massively interconnected. Each of these elements sends excitatory and inhibitory signals to other units that in turn update their behaviors based on these received messages. ANNs emulate the functionality of the human brain that implicitly stores knowledge in the interconnection weights, and not in the neurons themselves [13]. The learning process (i.e. knowledge acquisition) takes place by presenting *sample* cases to the network and modifying the connection weights between elements (through a learning algorithm) to model the representation of the cases [14]. Knowledge is stored in the final interconnections values. Because training examples need not be exhaustive, neural network methodology would ease the knowledge acquisition bottleneck that is hampering the creation of expert system [15]. The key to a neural network's power is the inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization. A typical network consists of a set of processing elements (PE) grouped hierarchically in layers and interconnected in some fashion. These PEs sum *N* weighted inputs and transfer that outcome through a mathematical function. Figure 1 shows a processing element and a three layer neural network. One of the most popular neural paradigms, and the one that has been used in this paper, is the multilayer perceptron trained using the Error Back Propagation Training Algorithm (EBPTA) [16]. Within the manufacturing environment ANNs have been applied to quality control, process control, part-family classification [17,18,19,20,21]. Neural network models have also been used for optimizing cutting parameters of turning operations [22].

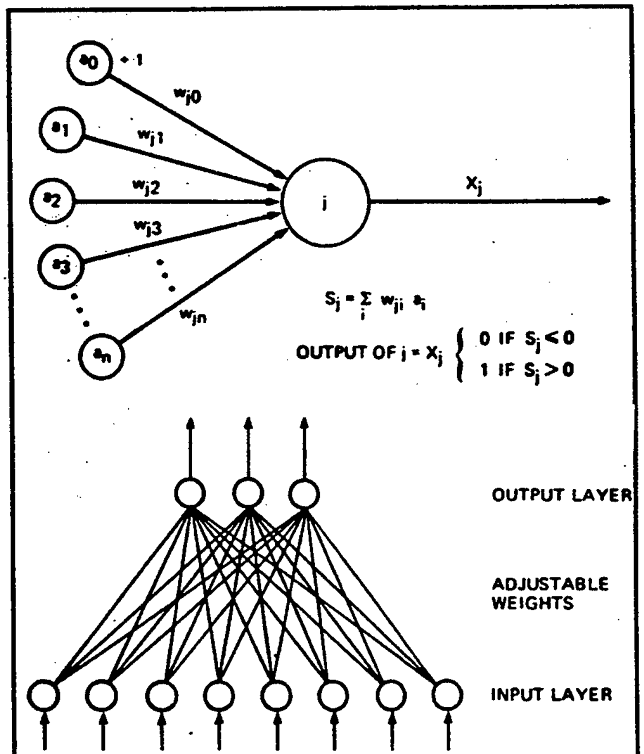

a_0
+1
w_{j0}
a_1
w_{j1}
a_2
w_{j2}
w_{j3}
a_3
a_n
w_{jn}
j
X_j
$S_j = \sum_i w_{ji}\, a_i$
OUTPUT OF $j = X_j \begin{cases} 0 \text{ IF } S_j < 0 \\ 1 \text{ IF } S_j > 0 \end{cases}$
OUTPUT LAYER
ADJUSTABLE WEIGHTS
INPUT LAYER

Figure 1. General Neural Network Components

3.0 PROCESS PLANNING: A PATTERN RECOGNITION PROBLEM

Pattern Recognition (PR) can be characterized as an information mapping process taking place over a set of (metric/non-metric) spaces [14]. An abstract view of the PR description problem is shown in Figure 2. A relationship G_i stands as a mapping between class-membership space, *C*, and a pattern space, *P*. Each Class w_i corresponds to a subset of patterns in the pattern space, where the i^{th} pattern is denoted as p_i. These pattern spaces may overlap allowing patterns from different classes to share the same attributes. Figure 2 shows mapping from the pattern space to an observation space, with feature patterns m_i, via the relationship *M*.

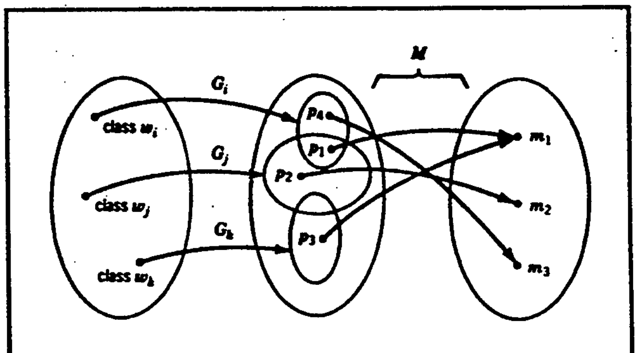


Figure 2. PR: mapping in an abstract form

All patterns pertaining to the different spaces are represented in an abstract form by vectors, and any given space may assume a specific dimension, *n*, and thus be represented generically as R^n. Depending on the nature of the problem (the pattern generating environment) and on how well it can be structured, a suitable PR technique can be applied. For example, statistical PR can be used when patterns come from a statistically characterized environment. However, quantitative precise descriptions of some complex decision problems may be largely unknown and a detailed characterization may be hopeless. A viable alternative is to treat the problem from an input/output or 'black box' viewpoint [14]. A Neural pattern recognizer is a good example of black box systems. Because process planning (PP) is such an ambiguous and ill-structured problem (at least in certain aspects), it seems reasonable to approach PP using neural network PR techniques.

Process planning can be cast as a pattern recognition problem; that is a set of mappings from one space(s) to another space(s). Process planning for CNC machining is a multivariable input/output process considering, among others, part dimensions, material, surface finish, fixture rigidity, tool material, spindle speed, feed rate, depth of cut, etc..

The input decision variables pertaining to the part to be manufactured are: feature type, feature perimeter, length, width, depth, corner radius, material, tolerance, and finish as shown in Figure 3. Other input decision variables are machinist preference and delivery requirement of the part. Each of the output decision variables depends on a mixture of the input variables, and possibly on some other input variables. The mixture of the input variables forms the input space, or in pattern recognition terminology the observation space. The output variables forms the output space, or likewise the decision space. The idea of the 'black box' is to collapsethe effects of individual input variables into a combined input effect, with no major emphasis on any particular input variable. The concept permits capturing inexact relationships between individual inputs and outputs. Figure 4 is a schematic view of a 'black box' process planner. Here, the output decision variables are the entry strategy, and cutting tool material.

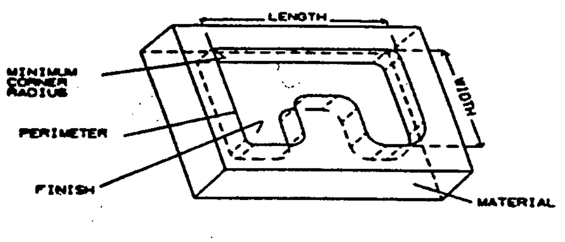


Figure 3. A Pocket Feature

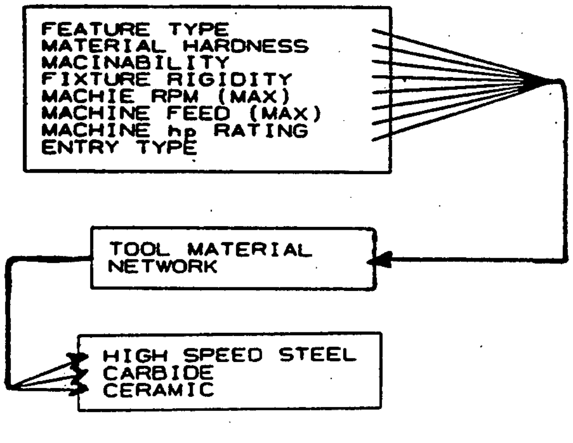


Figure 4. A Black Box View: Tool Type Selector

4.0 A NEURAL NETWORK PROCESS PLANNER

The research methodology adopted is aimed at exploring the utilization of neural network models as supportive decision-making tools in machining process planning and as a knowledge store. To carry out this objective, two neural network architectures have been implemented and tested to make decisions on tool type material and entry strategy for a CNC milling operation. For each network a necessary subset of milling process parameters have been identified as shown in Tables 1a, 1b.

Table 1a.

Network 1.

Output decision variable: **Tool material type**

Input variables:	Values (classes) of output:
Feature type	
Depth of cut	Carbide
Part material hardness	Ceramic
Machinability	High speed steel
Fixture rigidity	
M/C max rpm, feed, horsepower	

Table 1b.

Network 2.

Output decision variable: **Tool Entry strategy**

Input variables:	Values (classes) of output:
Feature type, # of Islands, smallest gap	
Feature Width, Depth, Length, Perimeter	Plunge
Part material hardness	Free entry
Fixture rigidity	Drill hole
M/C max rpm, feed, horsepower	Ramp
Machinability, Corner Radii, Surrounding geometry	

The development process of the network consists of two main steps. First, a preliminary data set was obtained for each decision variable considered, and an initial network architecture was selected for training to make sure that a neural model of the problem can be generated (for preliminary design, the number of hidden nodes was set equal half the input pattern dimension, along with a learning rate of .1 and a momentum factor of .9). Second, using a refined set of examples, a sensitivity analysis procedure was conducted to determine optimal network design parameters. The data collection phase was carried out at the Integrated Manufacturing Systems Laboratory at New Mexico State University from interviews with machinists. Forty five training examples were constructed for each decision output variable (i.e. about 10-15 examples for every output class) which provided a good level of discrimination between classes.

Two pre-processing steps took place before presenting data to the neural network. First, as shown in Table 1, input variables range in type and value. Some variables can intrinsically have numeric values (e.g. depth, hardness), while others (e.g. machinability and fixture rigidity) can best be described in linguistic form, such as high, medium or low. Since neural nets can only operate on numeric data, linguistic variables have been quantified so that they have numeric values between 0 and 1. Second, the values of variables can range from .0001's (for feed, tolerance) to 10k's (for machine rpm). During training of a network, a wide range of values can introduce a large amount of bias in the values of connection weights (i.e input variables of high values tend to be over-emphasized). To alleviate this problem, all input variables have been normalized so that their values fall in the range 0 and 1. This range is actually a preferable choice for network operation; as the activation function (unipolar sigmoid) assumes the same range, thus resulting in moderate values of the weights. When using the sigmoid activation function, a node cannot have an output of 1 or 0 unless a connection weight has an infinitely positive or negative value [23]. Therefore, .9 and .1 have been used as cutoff values for classification. During the learning phase, the network learns all possible classes and corresponding regions, and consequently, during the testing phase, the network recognizes a new input case (vector) by assigning it to the closest decision class. New cases are represented to the network as input vectors and the network will recommend the closest possible solution in terms of the output variable.

5.0 NETWORK TRAINING AND RESULTS

This section presents results of network analysis for the two networks under study; namely the tool material network and the entry strategy network. Throughout the analysis training was carried out using a two-layer network (one hidden node) with a momentum factor of .9, and convergence target error of 5% (that is, the output generated by the network was required to be within 5% of the desired values). To reduce the effect of random initial weights selected by the simulation program, each simulation run presented here is an average of two run repetitions. Performance measures were monitored for various values of the learning constant and number of hidden nodes.

For the tool material network, Table 2 shows the results of convergence cycles vs. number of hidden nodes at a learning rate of .1. Generally the number of convergence cycles increases with increase the number of hidden nodes. Hidden nodes provide the mechanism for representing the knowledge, and so it would be reasonable to include as many as possible without causing excessive or unnecessary computation. Thus, for this network the most appropriate number of hidden nodes is 16 as indicated in Table 2..

Table 2. Tool material network: convergence cycles vs. # of hidden nodes (η =.1)

network #	# of hidden neurons	# of converg. cycle
1.	5	1442
2.	8	2161
3.	11	1461
4.	13	1156
5.	16 *optimal	817
6.	20	1301
7.	25	1638

Table 3 shows the results of the convergence cycles vs. the learning rate. As expected there is a general increasing trend of the convergence cycles as the learning rate becomes smaller.

Table 3. Tool material network: convergence cycles vs. learning rate (hidden neurons=16)

learning constant(η)	convergence cycles
0.10	1034
0.20	513
0.35	691
0.40	577

A 0.2 learning rate was used as it provided the best convergence rate For the entry strategy network, the learning rate was varied for different values of the number of hidden nodes as shown in Table 4. The number of cycles decreased as the learning rate increased for different number of hidden nodes except where the number of hidden nodes is 16. This network did not converge at any learning rate when the number of hidden nodes was 25; it kept oscillating with an average error value of 61. So for this network the number of hidden used was 20 along with a 0.3 learning rate.

Table 4. Entry strategy network: convergence cycles vs. hidden nodes & learning constant

Hidden nodes	8	16	20	25
Learning constant				
.1	1973	1254	1357	*
.2	1425	890	749	*
.3	824	950	639	*

* did not converge

A second important performance measure of network operation is the percentage of classification errors. The networks designed were tested first, using the 45 training examples, and second, using a new set of 20 examples that were not used for training. For the tool material network, when tested using the training examples, the performance of the networks exceeded the expectations; as it was able to recognize at least 37 of the 40 training examples. This result supports the fact that a neural model *can* be synthesized to assist satisfactorily in the process planning decision-making process. When used to classify the new patterns, the network correct classification rate ranges from 69% for a learning constant of .30 and 16 hidden nodes to 92% for a learning constant of 0.10 and the same number of hidden nodes. These results are summarized in Table 5 as averages of two trials. Using this performance measure it seemed that a .1 learning rate is preferable to .2 and .3 as the average correct classification rate is the highest for a .1 learning rate. However, it is at the expense of increased convergence cycles and time (see Table 3).

Similar analysis was conducted for the entry strategy neural network. Again this network was able to correctly classify at least 90% of the examples used for training. When the network was tested for new examples the performance dropped down to a best value of 74% . The results of the correct classification percentage of the network trained with a 20 hidden nodes, .9 momentum factor, and 5% target .

CONCLUSION

This paper has shown the viability of utilizing the neural networks approach for assisting the process planner or CNC programmer in obtaining machining set-ups and strategies. It has been demonstrated that uncertainty and vagueness inherent in the minds of "experts" can be handled using a neural network approach within certain limits of error. The methodology shows a promise as a technique of storing knowledge and being retrievable in a rapid manner. As new manufacturing strategies become proven for industry, the neural network model can be easily retrained to handle updated information.

Table 5. Tool material network classification performance

hidden nodes	learning constant(η)	correct classification (%)	correct classification(%)
		Training Examples	Test Examples
5	.10	95	88
11	.10	95	90
16	.10	100	92
5	.20	90	80
11	.20	95	85
16	.20	97	90
5	.30	95	80
11	.30	95	76
16	.30	95	69

The optimal design of a neural network architecture is achieved through running simulations and in this case the "best" was sixteen hidden nodes and 0.1 learning factor for the tool material network, and 20 hidden nodes with a .1 learning rate for the entry strategy network, as measured by the percentage of correct classification. During the testing phase, the classification accuracy of the tool material network averaged about 80% correct classification over a range of 69% to 92%. However, the average classification accuracy for the entry strategy network ranged from 43% to 74%. These results should be encouraging as they provide evidence that a neural network model can represent some aspects of process planning decision-making.

At this stage, this pilot neural network process planner can enhance process planning by providing initial assistance in the decision process, either in a computer-aided or human-made decison-making environment. Future work is planned in constructing more training examples that are comprehensive and representative of the decision cases considered. It follows that these result arrays can be a good means of storing the knowledge in a retreivable format.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support received from Sandia National Labs for their support of initial technology transfer grants, to the Army Research Office for the present support and finally to students David Culler, John Holder and Eric Parsons for their work on the project.

BIBLIOGRAPHY

1. D. Bedworth, R. Henderon, M. Wolfe, *'Computer-Integrated Design and Manufacturing'*, McGraw-Hill, 1991.

2. L. Cox, D. Culler, and A. Al-Ghanim, *'Artificial Intelligence Application Utilization in Computer Numerical Control Manufacturing'*, in Proc. of IIE Conference, Los Angeles, May 26, 1993, pp(490-496).

3. Kusiak A., *'Group Technology for Flexible Manufacturing Systems'*, Handbook of Flexible Manufacturing Systems, Academic Press, San Diego, CA, 1991.

4. T. Gupta, *'A State of the Art Survey of Expert Systems in Process Planning'*, IIE Integrated System Conference in St. Louis, pp(269-274), Oct.30, 1988.

5. T. Gupta, *'A Survey of Expert Systems in Manufacturing and Process Planning: Current Development and it Future'*, Computers and Industrial Eng., Vol.18, No.1, pp(69-80), 1990.

6. L. Alting, H. Zhang, *'Computer-Aided Process Planning: state-of-the-art survey'*, International Journal of Production Research, Vol.27, No.1-4, pp(553-585), 1989.

7. Tsang J.P, and Brissuad D., *'A Feature-Based Approach to Process Planning'*, Computers in Industrial Engineering, 1989.

8. van Houten, van't Erve, and Kals, *'A feature Based Computer Aided Process Planning'*, Laboratorium voor Produktietechniek, Universiteit Twente, Netherlands, February 1989; in Proc. of the 21st CIRP International Seminar on Manufacturing Systems, Stockholm, 1989.

9. Y. Descote, J.C. Latombe, *'GARI: An Expert System for Process Planning, Solid Modeling Computers'*, Plenum Press, New York, 1983.

10. G. Jared, *'Shape Features in Geometric Modeling, Solid Modeling by Computers'*, Plenum Press, New York, 1983.

11. J. Giarratano, G. Riley, *'Experts Systems: principle and programming'*, PWS publishing company, Boston, 1989.

12. Amjed Al-Ghanim, and Bassam Talhouni, *'A Diagnostic Expert System for IBM PCs and Compatibles'*, Muta Journal for Scientific Research, Muata University, Jordan, (Accepted for publication).

13. Rumelhart D.E., McClelland J.L., and the PDP Research Group, *'Parallel Distributed Processing: Explorations in the Microstructure Cognition'*, Volume 1, MIT Press, 1986.

14. Shalkof R.J., *'Pattern Recognition: Statistical, Structural, and Neural Approaches'*, John Wiley & Sons, Inc., USA, 1992.

15. Zurada J.M., *'Introduction to Artificial Neural Systems'*, West Publishing Company, USA, 1992.

16. Dayhoff J.E., *'Neural Networks Architectures: An Introduction'*, van Nostrand International Company, New York, 1990.

17. S. Martinez, A. Smith, B. Bidanda *'A Neural Predictive Quality Model for Slip Casting Using Categorical Metrics'*, in Proc. of IIE Conf., Los Angeles, May 26, 1993, pp(265-269).

18. Tomas Telsaco, *'Interfacing Process Quality Control and CIM Through Application of Artificial Neural Networks'*, in Proc. of IIE Conference, Los Angeles, May 26, 1993, pp(280-284).

19. Godwin J. Udo, *'Neural Networks Applications in Manufacturing Processes'*, Computers and Industrial Eng. Vol.23, No.1-4,pp(97-100), 1992.

20. Deborah F. Cook, Robert F. Shannon, *'A predictive Neural Network Modeling System for Manufacturing Process Parameters'*, International Journal of Production Research, Vol.30, No.7, pp(1537-1550), 1992.

21. Kanda Charraprty, Utpal Roy, *'Connectionist Model for Part Family Classification'*, Computers and Industrial Eng., Vol.24, No.2, pp(189-198), 1993.

22. Jun Wang, *'A Neural Network Approach to Multiple-Objective Cutting Parameter Optimization Based on Fuzzy Preference Information'*, Computers and Industrial Engineering, Nos 1-4, pp(389-392), 1993.

23. Pao, Y.H., *'Adaptive Pattern Recognition and Neural Networks'*, Addison-Wesley, Mass, 1989.

Biographical Sketch

Leon D. Cox, Assoc. Prof of Industrial Engineering and Engineering Technology, New Mexico State University. Prof. Cox is the director of the Integrated Manufacturing Systems Laboratory and teaches courses in CIM, Quality Control and Production Systems. His research interests are in the area of manufacturing strategies.

Amjed M Al-Ghanim is a PhD candidate in Industrial Engineering at New Mexico state University. His research interests are in the areas of Artificial Intelligence and its applications to Manufacturing and Quality Control problems.

Pergamon

Computers ind. Engng Vol. 29, No. 1–4, pp. 217–220, 1995
Elsevier Science Ltd. Printed in Great Britain

0360-8352(95)00074-7

A Neural Network-Based Methodology for Machining Knowledge Acquisition

Leon D. Cox, Assoc.. Prof.
Engineering Technology and Industrial Engineering
Amjed M. Al-Ghanim, PhD
David E. Culler, PhD
Industrial Engineering Department
New Mexico State University

ABSTRACT

The paper presents results of a study on collecting machining strategies for machining assistants and process planning. These efforts are being conducted at the NMSU-Integrated Manufacturing Systems Laboratory (IMSL). Goals of the project aim at improving and advancing the solicitation, documentation, and automation of machining knowledge/data acquisition, and integration with CAD/CAM/CAE systems. This paper emphasizes the knowledge acquisition phase of the study utilizing artificial neural networks.

INTRODUCTION

Designing and developing knowledge based systems for Computer Numerical Control (CNC) machining operations is a formidable task due to the wide variety of part designs and the inability to anticipate local operating constraints. What works in one place does not necessarily work in another. The steep learning curves and high costs for implementing and using CAD/CAM has resulted in a very slow shift toward computerized machining. In order to cut lead times and reduce costs for operating CNC equipment, systems are needed which can be easily tuned and adjusted during implementation and then continuously improved.

Planning and developing Numerical Control (CNC) programs for the production of mechanical parts is an expensive and time-consuming process [1]. Part geometry, design specifications, local expertise and operating constraints form the basis for the selection of machining strategies which must be translated (by one of various methods) into specific machine controller instructions. A process engineer (domain expert), who traditionally performs these functions manually, routinely applies learned skills to prescribe manufacturing operational steps and specific machining parameters.

The IDEF3 (ICAM Definition Language 3) structured model of the thought process followed by a CNC programmer helps maintain a consistency throughout the machining assistants because it can be used to acquire and represent knowledge, format the decision processor, and assist the user in reviewing the reasons for suggested methods after a tool path is built. One main focus is to document, analyze, and classify various machining input parameters and techniques accumulated through literature review and interviews with experts.

Commercial CAD/CAM/CAPP software do not currently facilitate the automatic accumulation of design and manufacturing data produced throughout the planning and production of parts. Today's CNC machine tool programming methods are inefficient and do not utilize the product model concept. Plans are often produced and executed on the fly, modified on only the final hard copy, and inaccessible by people responsible for job costing. Furthermore, the intelligence and know-how to manufacture and track parts resides in wide base of skilled crafts people, process engineers and manufacturing experts. Advanced NC programming systems will be necessary to exercise the use of a product model. The Computer Aided Acquisition and Logistics Support (CALS) and Product Definition Exchange Specification (PDES) / Standard for the Exchange of Product Data (STEP) initiatives are currently addressing these very issues.

MACHINING STRATEGY

The feature being machined, available tooling, machine accuracy, and setup rigidity can influence one's decision drastically. Many times, duplicate tools are used to both rough and finish (especially when considering a large number of parts) so that results can be repeated consistently. Setup decisions are based upon the configuration of the part, the material and the tooling available. Setup choices for a particular feature can include tool selection, depth/width/cut, speed and feed.

Features. The concept of "features" plays an important role in linking design to manufacturing. CAM-I's Current Status of Features Technology defines features used in process planning for manufacturing as "shapes and technological attributes associated with manufacturing operations and tools [1]." From a machinist's viewpoint, the first step in developing a machining plan is to break the part into features. Each feature is characterized as a structural entity whose attributes specify lower-level geometry, topology, tolerance, surface finishes, etc.

Tool Selection. Roughing and Finishing tools are capable of cutting in a wide variety of materials and orientations. The tool selection process can be separated into a series of choices between materials and geometric attributes associated with grooving tools. The choices include a tool material, tool width, tool shape, tool length, and corner radius.

Deep Hole Drilling. Drill geometry is such that abrupt retraction out of a deep hole drilling process can result in tool breakage. To alleviate this problem, programming dwell for two spindle revolutions will let the chip thin to a zero thickness and allow a clean retraction. This type of knowledge is seldom incorporated in a standard drill cycle on a controller and must be added through programming mehods.

Etc. The knowledge parameters need to be evaluated by experts and incorporated into a corporate data base.

KNOWLEDGE ACQUISITION AND REPRESENTATION

A machinist in charge of designing a tailored process plan is always faced with the problem of selecting from a vast variety of possible combination values of machining parameters, such as speed, feed, depth of cut, tool material and size, tool entry strategy, cutting strategy,...etc. "It is very difficult to construct general, and yet simple, rules capable of handling all or most machining cases," said Joe Alejandrez, a 20-year-experience machinist form the Physical Science Laboratory (PSL). It is difficult to obtain the assistance of skilled machinists when attempting to build an automated programming system. Information is difficult to communicate verbally and formalize for the purpose of designing a knowledge base. A key factor in the effort lies in the re-training of shop-floor workers. If some decisions they normally make are being replaced by computer programs, then new knowledge must be given in return. The services of an experienced lathe operator (Ken Davis, 17 years between FreightLiner and BOEING) for two months, greatly contributed to the project success.

Automatic creation of NC programs requires; 1) user-interaction -- to gather location specific operating information, 2) interrogation of the features geometric description within the CAD database, 3) a file based communication link to a decision processor and, 4) An applications interface language capable of building tool paths from user generated programs. Very few commercial CAD/CAM systems provide the user with enough power to accomplish these tasks. Many CAPP systems have had to build their own primitive CAD system for input or used symbolic languages to output the parts geometric description. Obvious extensions to the automated process planner described here include automatic feature recognition, enhanced decision processing methods, part program optimization routines, and Group Technology (GT) coding schemes for individual rotational features. For this project we utilized ANVIL5000 CADD/CAM/CAE software for its application interface and integrated capabilities.

ARTIFICIAL NEURAL NETWORKS (ANNs)

ANNs are parallel distributed processing architectures composed of a large number of small interacting elements that are massively interconnected. Each of these elements sends excitatory and inhibitory signals to other units that in turn update their behaviors based on these received messages. ANNs emulate the functionality of the human brain that implicitly stores knowledge in the interconnection weights, and not in the neurons themselves [3]. A typical network consists of a set of processing elements (PE) grouped hierarchically in layers and interconnected in some fashion. These PEs sum N weighted inputs and transfer that outcome through a mathematical function. One of the most popular neural paradigms is the multilayer perceptron trained using the Error Back Propagation Training Algorithm (EBPTA) [2].

Neural network models have key advantages over rule-based systems. First, the learning process, or the knowledge acquisition process, takes place by presenting only sample machining cases to the network. Learning occurs through modifying the connection weights between network elements (by a learning algorithm) to model the representation of the cases [4]. Knowledge is thus stored implicitly in the final interconnections values. Second, because training examples need not be exhaustive, it is conceivable that neural network methodology could ease the knowledge acquisition bottleneck that is hampering the creation of rule-based systems [5]. Finally, when utilized as a decision processor, an artificial neural network has inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization.

A NEW KNOWLEDGE ACQUISITION METHODOLOGY

This methodology should satisfy a number of basic requirements. First, it must reduce the burden associated with developing rule-based systems by avoiding constructing general rules, because, as explained earlier, experts are not good at generalizing solutions. Second, this methodology must ensure that knowledge be cast in a form usable by neural systems; forms of rules are excluded. What is required by a neural system is a set of training examples, where each example represents a real machining case described by a vector of numeric values of process variables encoded in a convenient from. Third, consequently, the new methodology must employ effective instrumental knowledge gathering forms; and since no general inferential statements are required at this stage, the knowledge gathering forms can be mere data collection forms. The basic idea underlying this procedure is to enable the user to provide relevant training examples only, and the neural network, through the learning algorithm, will implicitly derive the IF--THEN rules.

Having decided on the output Process Planning (PP) decision variables to be automated, the next step was to find all related PP input variables that could affect the selection of the output variable. This process took place in an iterative manner as follows:

- Initially, a long list of possible input variables was constructed that could collectively drive the selection process of all output variables simultaneously.
- It was found that not all selected input variables do contribute to the selection of a given output decision value.
- This revisited understanding revealed that some of the input variables had to be deleted as they did not affect selecting the value of the output parameter. At this point, our forms were judged to provide 'good' instruments to start collecting data. Each form deals with one output variables, and only relevant non-redundant input variables.
- During data collection, our expert machinists tended to emphasize some input

variables in certain cases, while giving less weight to the same variables when dealing with other cases. Weights were assigned to variables according to 0-10 scale in an increasing order of merit.

- Using the data forms, a good number of examples were constructed (about 50) for output decision. At this stage, it was observed that some input variables constantly scored low weight rating (e.g. 2, 3). Those variables were deleted form the forms as they were considered to represent noise. This also reduced the dimensionality of the input vector.

In light of the above requirements, a neural-based machining knowledge acquisition technique is designed and implemented during the course of developing a neural model as a supportive decision making tool in machining process planning.

CONCLUSIONS

Computerized machining processes are slowly becoming commonplace in both large and small companies. Machine operators and programmers must be included in the design and development of such applications. Many systems are dependent on a skilled computer programmer to tune the knowledge and rules embedded in, what is often, very cryptic code. Creative inference techniques and clever knowledge structures cannot solve problems that are not well understood by the applications' designers. The straightforward use of data collection techniques that ANNs employ can be successfully used in intelligent machining assistants.

ACKNOWLEDGMENTS

The project is grateful to Army Research Office, Boeing Corp., Ladish Corp., Manufacturing and Consulting Services, Dornfeld Machine Co., Physical Sciences Laboratory and Sandia National Laboratories for their contributions of grants, expertise, equipment and software.

REFERENCES

[1] Burd, W.C., *The Milling Assistant,* SAND89-1519, Computer Aided Manufacturing Division, 7483, Sandia National Laboratories, Albuquerque, New Mexico, 1989.

[2] Dayhoff J.E., '*Neural Networks Architectures: An Introduction*', van Nostrand International Company, New York, 1990.

[3] Rumelhart D.E., McClelland J.L., and the PDP Research Group, '*Parallel Distributed Processing: Explorations in the Microstructure Cognition*', Volume 1, MIT Press, 1986.

[4] Shalkof R.J., '*Pattern Recognition: Statistical, Structural, and Neural Approaches*', John Wiley & Sons, Inc., USA, 1992.

[5] Zurada J.M., '*Introduction to Artificial Neural Systems*', West Publishing Company, USA 1992.

MACHINING STRATEGIES COLLECTED, STORED AND UTILIZED WITH ARTIFICIAL NEURAL NETWORKS

LEON D. COX, AMJED AL-GHANIM, & DAVID E. CULLER
New Mexico State University, Integrated Manufacturing Systems Lab
Box 30001 Dept. 4230, Las Cruces, NM 88003 (lcox@nmsu.edu)

ABSTRACT:
This paper presents results from an Army Research Office grant developing knowledge based systems for Computer Numerical Control (CNC) machining operations. One of the areas of development involves knowledge/data acquisition for input parameters to the knowledge processor. In our work we chose artificial neural networks as a mechanism for capturing data and then utilized the ability of the networks to determine certain process parameters for automating specific tasks in CNC programming. Artificial neural networks are well adapted to this task because they can be continuously improved.

INTRODUCTION

An increasing need for the rapid production of small batches and prototype parts will drive manufacturers to seek new ways to decrease lead times and costs associated with Computer Numerical Control (CNC) programming and production planning. Planning and developing CNC programs for the production of mechanical parts can be an expensive and time-consuming process. A process engineer (domain expert), who traditionally performs these functions routinely applies learned skills to prescribe manufacturing operation steps and specific machining parameters. One author summarizes the problem as follows:

> This knowledge is better described as many differently weighted pieces of advice than as a small set of constraints to be satisfied. Such planning involves taking into account a great variety of both technological rules and economic considerations. Furthermore, they are not absolute. They are more preferences, between which compromises may be necessary. In addition, they may differ somewhat from one company to another. In fact, they represent the experience and know-how of engineers. An immediate consequence is that the planning of machining sequences definitely does not have a unique solution. (Descotte and Latombe, 1989)

Problem Identification

Commercial CAD/CAM/CAPP software do not currently facilitate the automatic accumulation of design and manufacturing data produced throughout the planning and production of parts. Today's CNC machine tool programming methods are often inefficient and do not utilize the product model concept. Plans are often produced and executed on the fly, modified on only the final hard copy, and inaccessible by people responsible for job costing. Furthermore, the intelligence and know-how to manufacture and track parts resides in a wide base of skilled crafts-people, process engineers and manufacturing experts.

Commercial CAD/CAM gets limited acceptance among CNC machine operators and CNC programmers who perceive that their jobs are being replaced by computers. And of course the systems available for efficiently capturing the knowledge of the 'skilled' are for the most part still in a state of development.

Artificial Neural Networks

There has been an increase of interest in 'brain style computing' in terms of artificial neural networks (ANNs). ANNs are parallel distributed processing architectures composed of a huge number of small interacting elements that are massively interconnected. Each of these elements sends excitatory and inhibitory signals to other units that in turn update their behaviors based on these received messages. ANNs emulate the functionality of the human brain that implicitly stores knowledge in the interconnection weights, and not in the neurons themselves (Rumelhart and McClelland, 1986). Learning is thus achieved by modifying the connection weights between elements. Feedforward neural nets acquire knowledge through training, where by repeatedly presenting sample cases to the net, its interconnection weights change accordingly to model the representation of the cases (Shalkof, 1992). Knowledge is stored in the final interconnections values. As such neural network methodology would ease the knowledge acquisition bottleneck that is hampering the creation of expert system (Zurada, 1992). The key to a neural network's power is the inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization. A typical network consists of a set of processing elements (PE) grouped hierarchically in layers and interconnected in some fashion. These PEs sum N weighted inputs and transfer that outcome through a mathematical function. One of the most popular neural paradigms, and the one that has been used in this project, is the multilayer perceptron trained using the Error Back Propagation Training Algorithm (EBPTA) (Dayhoff, 1990). Within the manufacturing environment ANNs have been applied to quality control, process control, part-family classification (Martinez, Smith, and Bidanda, 1993; Telasco, 1993; Udo, 1992; Cook and Shannon, 1992; Charraprty and Uptal, 1993) Neural network models have been also used for optimizing cutting parameters of turning operations (Wang, 1993).

Neural network models have key advantages over rule-based systems. First, the learning process, or the knowledge acquisition process, takes place by presenting only sample (machining) cases to the network. Learning occurs through modifying the connection weights between network elements (by a learning algorithm) to model the representation of the cases (Shalkof, 1992). Knowledge is thus stored implicitly in the final interconnection values. Second, because training examples need not be exhaustive, it is conceivable that neural network methodology could ease the knowledge acquisition bottleneck that is hampering the creation of rule-based systems (Zurada, 1992). Finally, when utilized as a decision processor, an artificial neural network has inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization.

Computer Aided Process Planning

Computer Aided Process and operations Planning (CAPP) systems attempt to automate the decision making process necessary to map design specifications on the company specific manufacturing environment. Many CAPP systems utilize some form of Artificial Intelligence (AI) to model the human interaction previously required by the task. CAD/CAM software and databases are used to supply product descriptions and information management, respectively. Material removal operations provide a rich environment for the implementation of intelligent applications. Much of the current work in this field concentrates on the AI technique used to process the knowledge and/or the solid model design features utilized by the engineer to initially create the part (Anderson and Chang, 1990; Chung, Patel, and Cook, 1990; Joshi, Vissa, and Chang, 1989; Phillips and Mouleeswaran, 1985).

Manufacturing Assistants

Manufacturing Assistants are systems which emphasize the specific decisions and associated data which must be considered when selecting tools, cutting speeds, tool motions, etc. in a manufacturing or machining environment. As discussed here, the term machining is defined as *the removal of unwanted material from a workpiece in chip form so as to obtain a completed product that meets size, shape and finish characteristics specified by the designer.* The following factors contribute to the need for improved methods to manage the manufacture of machined parts: advanced machine tool and computer technology, higher quality and stricter tolerance and finish requirements, new materials such as plastics, composites, and exotic alloys, smaller batch sizes and prototype parts, more complex part descriptions, and a more competitive marketplace.

The Turning Assistant

The Turning Assistant (TA) is a direct outcome of an ARO project which automatically creates plans for Computer Numerical Control (CNC) lathe operations. Starting with a drawing and a set of predefined machines and cutting tools, the TA interactively defines and produces code for a variety of features such as profiles, grooves, and threads. The TA focuses on the formalization of domain knowledge, providing document support, and reducing the costs and lead times associated with the planning function. For advanced systems to be usable in shops which currently dominate small batch production of machined parts, flexible tools for fitting the application to the environment must be made available.

The Turning Assistant is a system that is based on previous work providing automated programming for a variety of milled features. The "Milling Assistant" (MA) (Burd, 1989) evolved from an Expert System based decision processor to a Case-Based Reasoning (CBR) system to automate the decision process required for machining milled and point-to-point features on CNC machine tools.

PRODUCT MODELS

The complete electronic product definition for a manufactured part consists of all the information related to designing, producing, inspecting, and managing the manufacture of that part. The data provided by the product model must be accessible, well organized, and conform to standards prevalent in both industry and government. The Computer

Aided Acquisition and Logistics Support (CALS) and Product Definition Exchange Specification (PDES) / Standard for the Exchange of Product Data (STEP) initiatives are currently addressing these very issues.

Knowledge processor

Numerous systems can be utilized for decision processing. The TA decision processor is written in GRAPL-IV, a FORTRAN 77 based application interface within the ANVIL-5000 CAD/CAM system and is used to determine machining strategies and effectively execute the operation. Another is based on a series of ANN's written in C, which can be called as needed from the application interface within the ANVIL-5000 CAD/CAM system. Another set of C programs can make calls to MetCAPP-III or MetCAPP-IV to acquire necessary machining parameters and cutting strategies.

KNOWLEDGE ACQUISITION AND REPRESENTATION

The most important and difficult aspect of this research is to develop effective tools with which to elicit knowledge from domain experts. The skills obtained through years of observation, training, improvising, and trial-and-error are not easy to explain verbally, document, or formalize for the purpose of building automated CNC programming systems. A significant amount of time must be spent in familiarizing oneself with the field and terminology before attempting to solicit strategies directly from the CNC programmers and machinists.

ANN's have a distinct advantage in speed, development and knowledge updating (hence learning). Automatic creation of CNC programs requires: 1) user-interaction -- to gather location specific operating information, 2) interrogation of the features geometric description within the CAD database, 3) a file based communication link to a decision processor and, 4) An applications interface language capable of building tool paths from user generated programs. Very few commercial CAD/CAM systems provide the user with enough power to accomplish these tasks.

Representing the Knowledge

This research helped to develop knowledge elicitation and representation tools that can be used to formalize the expertise gained through machining strategy meetings, literature review, and programmer feedback. It includes the design of questionnaires that can be used to document strategies in a set format during interviews or literature review and data collection forms for training ANN's (see Table 1).

TABLE 1: ANN TOOL TYPE DATA COLLECTION FORM

Input Decision Variables									Output Decision Variables
*	Variable Name	Value	Weight Factor						The correct tool material:
1	Part (stock) material		0	2	4	6	8	10	1. High Speed Steel
2	Part material bH		0	2	4	6	8	10	2. Carbide
3	Material Machinability		0	2	4	6	8	10	3. Ceramic
4	Feature Depth		0	2	4	6	8	10	Comments:

Artificial Neural Networks as Knowledge Acquisition Tools

The adopted research methodology aimed at investigating the utilization of a neural-based knowledge acquisition approach. It is based on the operational mode of neural systems, and taking advantage of our knowledge and experience in developing rule-based systems for similar tasks (i.e. milling assistant (Burd, 1990)). This methodology satisfies a number of basic requirements. First, it must reduce the burden associated with developing rule-based systems by avoiding constructing general rules, because, as explained earlier, experts are not good at generalizing solutions. Second, this methodology must ensure that knowledge be cast in a form usable by neural systems; forms of rules are excluded. What is required by a neural system is a set of training examples, where each example represents a real machining case described by a vector of numeric values of process variables encoded in a convenient from. Third, consequently, the new methodology must employ effective instrumental knowledge gathering forms; and since no general inferential statements are required at this stage, the knowledge gathering forms can be mere data collection forms

A neural network paradigm must first be selected in consensus with the application at hand. In this research, a *supervised* neural network model is employed as a *classifier*. A supervised learning model is selected as it enables the expert to express his judgmental opinion in a relative manner. This feature is very essential because, in this application, it is not required to group the training examples into clusters as would be the outcome if an *unsupervised* learning model were used. Instead, every single training example must be explicitly identified (by the expert) as belonging to a specific class that represents a machining decision. The neural model is employed as a classifier so that the output decision belongs to a class of objects (e.g. type of tool material: HSS, Carbide, Ceramic), or to a range of numeric values (e.g. tool size: 1/4"-1/2", 1/2"-1", etc.). This method shows great promise considering the variety of machining choices that can satisfy the requirements of a quality product. See Table 2 for a representative example of the ANN training and testing results.

ACKNOWLEDGMENTS

The members of this project are grateful to Army Research Office, Boeing Corp., Ladish Corp., Manufacturing and Consulting Services, Institute for Advanced Manufacturing Sciences, Physical Sciences Laboratory and Sandia National Laboratories for their contributions of grants, expertise, equipment and software.

TABLE 2: TOOL MATERIAL NETWORK CLASSIFICATION PERFORMANCE

Training Examples			Testing Examples		
hidden nodes	learning constant(η)	identification correct (%)	hidden nodes	learning constant(η)	identification correct (%)
5	.20	90	5	.20	80
11	.20	95	11	.20	85
16	.20	97	16	.20	90

REFERENCES

Anderson, D.C., and Chang, T.C., (1990). Geometric reasoning in feature-based design and process planning. *Comput. & Graphics*, Vol. 14, No. 2, pp. 225-235.

Burd, W.C., (1989). The Milling Assistant, *SAND89-1519, Computer Aided Manufacturing Division*, 7483, Sandia National Laboratories, Albuquerque, New Mexico, 1989.

Charraprty, Kanda, and Utpal, Roy, (1993). Connectionist model for part family classification, *Computers and Industrial Eng.*, Vol.24, No.2, pp.189-198.

Chung, J.C.H., Patel, D.R., & Cook, R.L., (1990). Feature-based modeling for mechanical design, *Comput. & Graphics*, Vol. 14, No. 2, pp. 189-199.

Cook, Deborah F., and Shannon, Robert F., (1992). A predictive neural network modeling system for manufacturing process parameters, *International Journal of Production Research*, Vol.30, No.7, pp.1537-1550.

Dayhoff J.E., (1990) *Neural networks architectures: An introduction*, van Nostrand International Company, New York.

Descotte, Y., and Latombe, J.C., (1989). Gari: A problem solver that plans how to machine mechanical parts, *I.M.A.G.*, Grenoble Cedex, France.

Gallant, Stephen, (1993). Neural Network Learning and Expert Systems, MIT press.

Joshi, Vissa, & Chang, (1989). Expert process planning system with a solid model interface, Chapter 6, *Knowledge-Based Systems in Manufacturing*, Taylor and Francis Inc. Philadelphia, PA.

Martinez, S., Smith, A., Bidanda, B. (1993). A neural predictive quality model for slip casting using categorical metrics, *Proc. of IIE Conf.*, Los Angeles, May 26, 1993, pp.265-269.

Phillips, R.H., and Mouleeswaran, C.B., (1985). A knowledge-based approach to generative process planning, *CASA/SME Autofact 85 Conference*, November 1985.

Rumelhart D.E., McClelland J.L., & the PDP Research Group, (1986). *Parallel distributed processing: Explorations in the microstructure cognition*, Volume 1, MIT Press.

Shalkof R.J., (1992). *Pattern recognition: Statistical, structural, and neural approaches*, John Wiley & Sons, Inc., USA.

Telsaco, Thomas, (1993). Interfacing process quality control and CIM through application of artificial neural networks, *Proc. of IIE Conference*, Los Angeles, May 26, 1993, pp.280-284.

Udo, Godwin J., (1992). Neural networks applications in manufacturing processes, *Computers and Industrial Eng.*, Vol.23, No.1-4, pp.97-100.

Wang, Jun, (1993). A neural network approach to multiple-objective cutting parameter optimization based on fuzzy preference information, *Computers and Industrial Engineering*, Nos 1-4, pp.389-392.

Zurada J.M., (1992) *Introduction to artificial neural systems*, West Publishing Company, USA.

A Process Planning Technique Utilizing Neural Networks to Determine CNC Milling, Drilling and Turning Parameters and Strategies

ABSTRACT

As related to machining, process planning consists of a series of tasks including interpretation of product model, selection of machine tools, selection of tool sets, selection of setups, design of fixtures, selection of machining sequences, determination of tool paths, calculation of machining parameter, and generation of NC programs. Process planning is an involved activity that has a large number of input variables describing the part to be produced and the production resources.

If a product model existed that contained all processing parameters necessary for the manufacture of the product then a subset of these would contain information that would need to take place during the process plan development. Virtually all products require a chip producing machine tool in some stage of manufacture and much of the chip producing is being accomplished through computer numerical control (CNC).

In most cases during the planning for the CNC machining operations it is necessary to consult an expert. Whether that expert is a machinist, part programmer or a knowledge base, the determination of specific set-up and run parameters or for strategies for the machining process must be known before a good part program can be completed. In some cases this is simply human stored knowledge and in others it is the additional inputs to the human stored knowledge. In our work we are focusing on the ability of the human (and in the long run the ability of a pattern recognition system) to identify a feature that needs to be machined such as a pocket, notch, slot, hole, shoulder, knurl, etc. Once identified that feature will have parameters and strategies developed specific to its nature, thereby, taking the larger problem and simplifying it.

Most research efforts have focused on utilizing expert systems technology to implement process planning systems. These systems offer assistance based on capabilities ranging from fast retrieval of existing plans to generating plans based on interpreting product model geometry. However, expert systems, due to explicit knowledge representation, show limited adaptive capabilities, and they do not tolerate missing or inaccurate data.

To compensate for some of these limitations and enhance the expert systems approach, this paper describes a methodology of process planning for CNC milling, drilling and turning based on the concept of artificial neural networks.

First, process planning has been cast as a pattern recognition problem and viewed as a mapping function between design features and machining features. Second, a neural network model is developed for automating process planning decisions such as cutting tool material, entry method, and cutting strategies. The viability of this approach stems from the intrinsic ability of neural nets to handle ill-structured problems, and their striking ability of generalization; an essential property that does not smoothly lend itself to existing approaches like expert systems (ES). Based on simulation results, initial investigation applied to milling processes with different features, promise a high potential for the development of a neural network-based process planning system.

IE 528
Kathryn Lueders
Fall 1995

Abstract

The paper presents results of an effort to utilize a commercially available process planning database to provide machining knowledge as input into a commercial CAD/CAM package. A prototype system was developed which utilizes the CAD/CAM software, a knowledge base and the process planning package through an application interface. The prototype system based upon a developed CAD drawing and basic initial decisions provides machining strategies through either the knowledge base or the process planning package. The goals of the project were to demonstrate the successful utilization of a knowledge base, a commercial process planning tool, and a neural net for inputs into a developed machining expert system.

Background

Process planning is defined by the Society of Manufacturing engineers as "the systematic determination of the methods by which a product is to be manufactured economically and competitively".[1] As the average age of aerospace process planners has been estimated by private industry and the Air Force to be 51 to 55 years. [2] The replacements for these planners do not have the level of expertise and knowledge to replace what will be lost through retirement. But even more importantly, computerized CAPP systems capture the expertise so that the knowledge is not vulnerable to attrition. The computer's ability to store, retrieve, and update data en masse makes it the ideal tool to organize the technological data representing the knowledge of senior process engineers in a useable format.

CAPP applications can be enhanced in three areas, CAPP retrieval, the ability of CAPP systems to interface with other business systems and enhanced communications.[3] Process planning may capture process information and render it useful through one of tow means: variant and generative process planning. Variant process planning relies on a human operator and a retrieval mechanism to retrieve similar process plans and edit them producing a new variation of a preexisting process plan: hence, the name variant process planning. Generative process planning represent knowledge in the form of

[1] pg. 8, Culler, David, "The Turning Assistant: Automated Planning for Numerical Control Lathe Operations", 1994.

[2] pg. 121, Nolen, James, Computer-Automated Process Planning for World-Class Manufacturing. Marcel Dekker, NY, 1989.

[3] pg. 182, Nolen

processing rules. These rules, often represented by decision trees, represent the collective manufacturing knowledge for a particular part family or category of manufacturing processes. Once captured, they may be used to generate the required sequence of processing steps for new products within the set of products for which the decision tree was designed.

Generative process plans have a number of advantages. Among the major ones are the following:

- They rely less on group technology code numbers since the process log, usually captured in a decision tree, often categorizes parts into families.
- Maintenance and updating of stored process plans are largely unnecessary since any plan may be quickly regenerated by processing through the tree. Indeed, many argue that with generative systems, process plans should not be stored since if the process is changed, an out-to-date process plan might find its way back into the system.
- The process logic rules, however, must be maintained up to date and ready for use. This provides the process planner with the assurance that the processes generated will reflect state-of-the-art technology.[4]

Knowledge Based Systems

The NMSU Integrated Manufacturing Systems Laboratory (IMSL) demonstrated the feasibility of utilizing expert or knowledge based systems in their generative process planning machining assistants. An expert or knowledge based system is described as consisting of a mass of information on a particular subject (the knowledge base) and a program which permits the system to apply its knowledge to a particular problem. The knowledge base consists of facts and rules. The facts constitute the body of information that is widely shared, publicly available, and generally agreed upon by experts in the field. The rules are rules of good judgment gleaned from a human domain expert that characterize expert level decision making in the field.

The Milling Assistant, initially developed by Bill Burd of Sandia National Labs, aids in the programming of Computer Numerical Control (CNC) machine tools by automating the decision process used to create cutter paths. MA is made up of a user interface to a commercial CAD/CAM system, ANVIL5K, analysis programs written in GRAPL and FORTRAN, and a decision processor which contains the rules and strategies on machining features. The MA utilizes a Case Base Reasoning (CBR) system for making strategic decisions. The user, based upon an existing part geometry, selects from the available cutting tools, material, and machine tools available in the MA. The MA, using those inputs, generates an effective tool path and then interacts further to allow the user to make desired modifications to the proposed solution.

[4] pg. 161, Nolen

Another IMSL product, The Turning Assistant (TA) developed by David Culler, based upon the MA model is a feature based process planner for numerical control lathe operations. Again, a Fortran 77 based decision processor is used to determine the machining strategies. The decision processor in both of these assistants utilizes user inputs of tools, stock, machine, and features and outputs suggested cutting tools, speeds and feeds, cut amounts, suggested entry points and patterns. Through the development of these programs, IMSL staff personnel learned how time consuming and difficult it was to develop a knowledge base which when interrogated appropriately provides "good" production plans.

Artificial Neural Networks

With the increasing level of complexity and an increasing level of complexity of manufacturing environments, and a further increasing demand for integration, maintenance of the expert system becomes difficult. In today's dynamic manufacturing environment, the expert system shows some weakness in its difficulty with being quickly modified or upgraded. The efficiency and accuracy of the decision processor is in question if it becomes cumbersome to program in the various decision rules required to provide the best solution.[5]

It was recognized by the IMSL personnel that designing and developing knowledge based systems for CNC machining operations could be formidable based upon the wide variety of part designs and the inability to anticipate local operating constraints. Further research was done to investigate methods of cutting the lead times and costs for developing process planners by either implementing a system which can be easily fine tuned and adjusted. In the pursuit of this goal, the IMSL project staff studied a methodology of process planning for CNC milling based on the concept of artificial neural networks (ANN). [6]

ANNs are parallel distributed processing architectures composed of a huge number of small interacting elements that are massively interconnected. Each of these elements sends excitatory and inhibitory signals to other units that in turn update their behaviors based on these received messages. Basically, ANNs are built to imitate the human brain. Learning is done by modifying the weights of the connections between elements. Several advantages were discovered by using the neural net over knowledge based systems. The first was that the neural net was "taught" by presenting sample cases with solutions. The net based on the inputs and desired outputs, through a learning algorithm, develops the relationships necessary to provide the desired outcome when insufficient parameters, i.e. inputs without an output, are introduced to the net. Secondly because training examples do not need to be exhaustive, it would seem to be an

[5] A. Al-Ghanim, L. Cox, "A Process Planning Technique Utilizing Neural networks to Determine CNC Milling Parameters and Strategies", 1994.

[6] A. Al-Ghanim, L. Cox, "A Neural Network Approach for Computer-Aided Process Planning of CNC Milling Parameters and Strategies.

improvement on the knowledge acquisition bottleneck often seen in the development or modification of knowledge bases.

The IMSL process planning neural net application was developed by Amjed Al-Ghanim in two phases. First, process planning was cast as a pattern recognition problem, and viewed as a mapping function between design features and machining features. Second, a neural network model was developed for automating certain aspects of process planning decisions such as the cutting tool material. Iterative analysis of this model served two objectives. First, the neural net venture reduced the burden of constructing general rules into a simple data collection process. For the sample models to be developed, effective machining data collection forms were designed and utilized. Second, the model based on the sample cases was proven capable of making suitable machining decisions.[7]

The neural net research also highlighted some weaknesses. The integrity of the neural net is directly related to the integrity of the sample cases. The expert machinist sometimes inconsistently weighted identical variables from case to case. By modifying the input form these inconsistencies could be minimized. Also, the neural net could provide inconsistent responses from day to day based upon the user.

Commercial Manufacturing Software Interfaces

Both the knowledge base and the neural net applications showed promise for the future but IMSL personnel pursued the ability of CAPP systems to interface with other software. This work was first begun by Gordon Hunt when he developed an interface between the expert system database already developed by the commercial process planning package MetCAPP and the CAD/CAM package ANVIL5K. The operation type, work piece material, and tool class were provided to the MetCAPP Application Programming Interface (API) and the depth of cut, feeds and speeds are then returned to ANVIL5K and input into the appropriate CAM modal settings which are used in the development of tool paths. The interface was expanded on with an integrated application between MetCAPP and the Turning Assistant. The option was added to the TA to access MetCAPPs already extensive database of materials and tooling and utilize that for some basic machining decisions. The MetCAPP's interface provides access to a larger knowledge base consisting of an extensive material and tooling database, and a mature expert machining system but the TA provides the expertise for interfacing with the ANVIL5K CAD and CAM package and the necessary creation of the tool paths that are the necessary first step toward NG code.

The utilization of the strengths of both packages is a true application of integrated manufacturing systems and hopefully, a look into the future of CIM systems. The greatest problem facing computer integrated manufacturing has been the inability of

[7] A. Al-Ghanim and L. Cox, "A Process Planning Technique..."

CAD/CAM and CAPP computer packages to interface with each other.[8] The latest application showed the feasibility of it if the packages provide the interfaces. The creators of MetCAPP and ANVIL understood that requirement and provided the interfaces for the user to customize the application and/or link applications together.

[8] J. de Vries, O.W. Salomons, A.H. Streppel, L.J. de Vin & H.J.J. Kals, "CAD-CAPP Integration for Sheet Metal Products"

References

A. Al-Ghanim and L. Cox, "A Process Planning Technique Utilizing Neural Networks to Determine CNC Milling Parameters and Strategies".

A. Al-Ghanim and L. Cox, "A Neural Network Approach for Computer-Aided Process Planning of CNC Milling Parameters and Strategies".

Burd, W.C., "The Milling Assistant", SAND89-1519, Computer Aided Manufacturing Division, 7483, Sandia National Labs.

D. Culler, "The Turning Assistant: Automated Planning for Numerical Control Lathe Operations"

L. Cox, D. Culler, and A. Al-Ghanim, "Artificial Intelligence Application Utilization in Computer Numerical Control Machining".

G. Hunt, "Machining Data Exchange Between Anvil 5K and MetCAPP".

Nolen, James, Computer-Automated Process Planning for World-Class Manufacturing, Marcel Dekker, NY, 1989.

J. de Vries, O.W. Salomons, A.H. Streppel, L.J. de Vin & H.J.J. Kals, "CAD-CAPP Integration for Sheet Metal Products"

Kathryn Lueders
December, 1995
Documentation of MetCAPP/TA Interface

I. Objective

The objective was to prove the feasibility of utilizing an interface between a commercial process planning database, in this case MetCAPP and the Turning Assistant, a NMSU Integrated Manufacturing Systems Laboratory (IMSL) developed integrated feature based CAD/CAM process planning package. The Turning Assistant previously used a developed knowledge base to determine the appropriate speeds and feeds based upon the type of machining operation, the part's material type and the tool class. Yet, the knowledge base was limited to the developer's expert knowledge. The MetCAPP database in comparison is a vast resource, already developed and externally maintained.

II. Background

The Turning Assistant (TA) is an NMSU Integrated Manufacturing Systems Laboratory (IMSL) developed project based upon the another IMSL project, the Milling Assistant (MA). The TA did for turning operations what the MA performed for milling operations, it took a completed CAD drawing and based upon user inputs used a case based reasoning system to automate the decision process necessary to develop NC tool paths. Both of these packages used a GRAPL IV application interface to access the knowledge base. Utilizing the user's inputs, the appropriate machining operation decisions are then passed back to the CAD/CAM package ANVIL5K, where system parameters are then set and tool paths based on those parameters developed. These tool paths are output as .clp files, and through a post processor utilized to develop the NG code that directs the machining of the part.

MetCAPP is an established commercial process planning package. It is made up of CUTPLAN, a process planning module, CUTDATA, the machining parameter information database, and CUTTECH, the MetCAPP's expert machining system. All of these integrated pieces provide the planner with data on the time and cost impacts of various machining decisions. For example, through CUTDATA, recommendations are provided on cutter diameter, speed, feed, tool material, machine rpm and ipm, and power based on user inputs of operation, material and tool class. These can duly be input into a CUTPLAN matrix where these inputs are translated into cutting time, labor hours, material used, and machine hours. What MetCAPP does not do is provide these inputs from a feature based model.

III. Scope

A menu was added to the TA where the user was queried as to whether they wanted to utilize MetCAPP. The user was then prompted for the required inputs of material type, machining operation, and tool type. These inputs were passed to TA building on an interface developed by Gordon Hunt. Hunt initially developed an interface between ANVIL5K and METCAPP that extracted base speed and feed data based upon the initial inputs. These outputs were in a format which with a conversion could be used as the necessary machine speeds and feeds required by Anvil's CAM package. If the user utilized the MetCAPP option for obtaining speed and feed data, the existing knowledge base resident in the TA was not accessed for that specific information.

IV. Methods

The NC_Turn was modified to incorporate a CALL statement to machdata.grs. Machdata.grs was the top tier program that then called:

- subroutine 'submat' for user definition of the part material.
- subroutine 'subtool.grs' for user definition of the type of tooling.
- subroutine 'subop.grs' for user definition of the type of operation. These were limited to turning operations since the application was being utilized in the TA.

The operation, tool type and operation are then passed through to 'getcapp' via a XCALL function. 'Getcapp' is a Standard C program that opens communication between the MetCAPP CUTDATA database and the function 'cappdata.c.' 'Cappdata.c' controls the functions that utilize the MetCAPP Application Programming Interface to create a new session of CUTDATA and pulls the required data out and stores it in various buffers as matches for key files are found. Cappstor.txt is then opened to write the operational inputs and the outputs to a file. If no match is found then an error flag is returned. The program ends and control is again returned to 'getcapp.'

A function called 'dataout' is initiated which opens cappstor.txt and transforms the character data to either string, integer, or double precision is performed on every variable read from the file. The GRAPL-IV program machdata.grs then receives the data back. The modals are set with the new parameters for speeds and feeds. Within machdata.grs a flag is set to indicate that the selection of material and tools has already occurred. Control of the program is given back to NC_Turn and the tool paths are selected based upon the MetCAPP inputs.

Turning Assistant Changes

The changes to the Turning Assistant included:

- A query and menus for selection of turning operation, part material, and type of tool.
- Flag check to see if the above selections were done prior to entering the TA decision processor.
- Added a conversion factor to correct the MetCAPP inputs to ipm or rpm instead of ifm as his program input them.
- Added a subroutine to verify the machine tool limitations with the MetCAPP inputs. The MetCAPP expert system only takes into account the type of process and the part material in its calculations of the various speeds and feeds. This caused problem when the machine tool's capabilities were the limiting factor to the speeds and feeds. The TA was modified to make a comparison between the MetCAPP inputs and the TA's tool database so that the modal change only incorporated the limiting speed or feed.

V. Conclusion

The success of the interface between the CAD/CAM package, a CAPP system and a knowledge based processor is a true example of the potential for integrated manufacturing software. Only recently have manufacturing software packages even made the attempt to provide interfaces or the easy customization due to the extra money in performing the customization or forcing users to buy large expensive packages. Therefore, MetCAPP and Anvil 5K are unique by providing application interfaces that allow the user to expand their capabilities and customize their use of the software.

The utilization of the strengths of both packages is a true application of integrated manufacturing systems and hopefully, a look into the future of CIM systems. The greatest problem facing computer integrated manufacturing has been the inability of CAD/CAM and CAPP computer packages to interface with each other.[1] This application showed the feasibility of it if the packages provide the interfaces.

[1] J. de Vries, O.W. Salomons, A.H. Streppel, L.J. de Vin & H.J.J. Kals, "CAD-CAPP Integration for Sheet Metal Products"

KNOWLEDGE ACQUISITION TECHNIQUES FOR INTELLIGENT MACHINING ASSISTANTS

Leon D. Cox, Assc. Prof.
Engineering Technology and Industrial Engineering
Amjed M. Al-Ghanim, PhD
David E. Culler, PhD
Industrial Engineering Department
New Mexico State University

ABSTRACT

This paper presents analysis and results of implementing two machining knowledge acquisition methodologies. The study is part of the machining strategy, machining assistant and process planning research efforts being conducted at NMSU-Integrated Manufacturing Systems Laboratory (IMSL). Goals of the project aim at improving and advancing the solicitation, documentation, and automation of machining knowledge. To achieve these goals, this research project has three major interrelated phases through which the intended work has been conducted. These phases are machining knowledge/data acquisition, automation, and integration with CAD/CAM/CAE systems. This paper emphasizes the knowledge acquisition phase of the study.

Designing and developing knowledge based systems for Computer Numerical Control (CNC) machining operations is a very formidable task due to the wide variety of part designs and the inability to anticipate local operating constraints. What works in one place does not necessarily work in another. The steep learning curves and high costs for implementing and using CAD/CAM has resulted in a very slow shift toward computerized machining. In order to cut lead times and reduce costs for operating CNC equipment, systems are needed which can be easily tuned and adjusted during implementation and then continuously improved.

KEY WORDS: (CAD/CAM, CAPP, CNC, Machining Strategy, Knowledge Base)

INTRODUCTION

Planning and developing Numerical Control (CNC) programs for the production of mechanical parts is an expensive and time-consuming process. Part geometry, design specifications, local expertise and operating constraints form the basis for the selection machining strategies which must be translated (by one of various methods) into specific machine controller instructions. A process engineer (domain expert), who traditionally performs these functions manually, routinely applies learned skills to prescribe manufacturing operational steps and specific machining parameters.

In one investigation the project staff developed a prototype neural network model that could handle few Process Planning (PP) functions[2]. Iterative analysis of this model served two objectives. First, the development process of this pilot system provided a new innovative machining knowledge acquisition methodology. This neural-based venture reduced the burden of constructing general rules into a simple data collection process. Our efforts resulted in designing and utilizing a set of effective instrumental machining data collection forms. And, second, the model provided a promising neural-based PP system capable of automating suitable planning functions.

One main focus is to document, analyze, and classify various machining input parameters and techniques accumulated through literature review and interviews with experts. The IDEF3 (ICAM Definition Language 3) structured model of the thought process followed by a CNC programmer helps maintain a consistency throughout the machining assistants because it can be used to acquire and represent knowledge, format the decision processor, and assist the user in reviewing the reasons for suggested methods after a tool path is built.

Commercial CAD/CAM/CAPP software do not currently facilitate the automatic accumulation of design and manufacturing data produced throughout the planning and production of parts. Today's CNC machine tool programming methods are inefficient and do not utilize the product model concept. Plans are often produced and executed on the fly, modified on only the final hard copy, and inaccessible by people responsible for job costing. Furthermore, the intelligence and know-how to manufacture and track parts resides in wide base of skilled craftsmen (and women), process engineers and manufacturing experts. Advanced NC programming systems will be necessary to exercise the use of a product model.

Machine tool manufacturers have benefitted greatly over the years from customer feedback. Shared techniques, machinability tests, and performance evaluations have educated and diversified them in the areas of computer hardware and CNC. Many machines have powerful built in programming systems that compete closely with the option of CAD/CAM. The computer hardware, software, and communication links built in as part of the machine tend to be primitive. The problem is that these machines and controller interfaces end up being the place where all process planning occurs. One term used to describe it is "programming at the spindle." Operations can become dependent on a certain machine/operator and programs or parts cannot be re-routed in case of breakdowns or scheduling problems. There is no intermediate "generic" cutter location data file which can be postprocessed for an alternative machine with similar capabilities. Furthermore, there is no level of control past the machine interface. Design, planning, manufacturing, and inspection data is difficult to link when so much of it resides in the relatively inaccessible machine controller unit. Operators traditionally are very good verbal communicators but have limited writing and documentation training. Commercial CAD/CAM gets limited acceptance among operators/NC programmers who perceive that their jobs are being replaced by computers.

Process Planning. Process Planning (PP) is defined by the society of manufacturing engineers as 'the systematic determination of the methods by which a product is to be manufactured economically and competitively'. The task of process planning involves a series of steps, the first of which is the interpretation of the product design data which consists of geometric configuration, dimensions and tolerances, and raw material properties. Second, machine tools capable of performing the required processes will have to be selected based on machine tools availability, capability, range of machining operations, required production rates, etc.. The operation sequence is then determined based on groups of parts to be produced. The tool sets, jigs and fixtures, and cutting conditions such as the depth of cut, feed rate, speed of cutting tool have to be determined. Finally process sheets (route sheets) and part programs are developed, taking into consideration the aforementioned factors [2]. In summary, process planning for machining can be itemized as a sequence of:

- Interpretation of product design data.
- Selection of machining processes, machines tools, and cutters.
- Determination and design of jigs and fixtures.
- Sequencing of the selected operations.
- Determination of proper cutting conditions.
- Generation of process sheets and numerical control (NC) part programs (in the case of automated manufacturing).

Thus, a process planner is a manufacturing subsystem responsible for converting design data into work instructions. It represents the link between engineering design and shop floor manufacturing, and determines the manufacturing operations required to transform a part from a rough state to a finished state specified by the engineering drawing. In discrete-part metal-cutting operations (e.g. milling, turning, grinding, drilling..etc.), a number of computational procedures (e.g. mathematical models, heuristics) is required to determine the above itemized macro-level tasks and associated micro-level decision variables. For example, based on the part identifiable manufacturing features (or machining surfaces), required finish and tolerances, production rate, and available machine capabilities, an appropriate (perhaps optimal) production process(s) can be selected to machine that feature. Once a process is selected then utilizing such variables as part material characteristics (e.g. hardness, machinability), part dimensions, cost of raw material, cutting cost, a set of cutting conditions like cutting speed, feed rate, depth of cut, number of passes, and usage of coolants can be determined.

Process Planning Automation. Computer Aided Process and operations Planning (CAPP) systems attempt to automate the decision making process necessary to map design specifications on the company specific manufacturing environment. Many CAPP systems utilize some form of Artificial Intelligence (AI) to model the human interaction previously required by the task. CAD/CAM software and databases are used to supply product descriptions and information management, respectively. Material removal operations provide a rich environment for the implementation of intelligent applications. Much of the current work in this field concentrates on the AI technique used to process the knowledge and/or the solid model design features utilized by the engineer to initially create the part [1,5,7,8].

Process planning is a difficult task to automate because it comprises of a series of functions that do not submit 'nicely' to modeling techniques. Nevertheless, there are two well-known systematic computer-aided process planning (CAPP) strategies, namely, *variant planning* and *generative planning*. The distinguishing feature of the variant planning strategy is that former plans are retrieved and modified for new parts, while generative planning is a strategy that strives to create new part plans from scratch based on analyzing part geometry and other related specifications [4]. A hybrid of the variant and generative is common.

Developing knowledge acquisition and representation tools which allow information pertaining to specific features to be documented and evaluated is viewed as an important part of a production-worthy CAPP system. Generative AI programming languages all require the same knowledge as the basis for structuring a decision processor, regardless of the technique. Many CAPP systems require AI programmers to modify and enhance the rule bases. Breaking the decision process down into small steps allows for a procedural approach to the actual decision processing. Knowledge attained during meetings with experts along with the resulting rules formulated for that particular decision can be stored in the model.

Process Flow Model. In 1981, the U.S. Air Force Integrated Computer-Aided Manufacturing (ICAM) program developed structured methods for applying computer technology to manufacturing. These methods were designed to better understand how to best improve manufacturing productivity. There are three well-documented modeling methodologies defined and developed around the ICAM Definition (IDEF) approach to system study; IDEF0, IDEF1, and IDEF2.

The Air Force contractor responsible for the continued development and maintenance of the public domain IDEF system/software definition, design, and engineering methods is Knowledge Based Systems (KBS) Inc. (College Station TX). KBSI has expanded the original IDEF languages to include IDEF3-IDEF6.

The IDEF methodology can be applied as a descriptive tool or an analysis tool. The descriptive tool uses heuristic models and is applied to complex problems such as manufacturing processes. The analysis tool requires a more in-depth evaluation and is applied to more narrowly defined problems such as business operations or practices. The idea is to establish a common language to enhance communications and evaluation and to stimulate organizational learning. It uses a graphical approach to describe the manufacturing functions of a system. A standard notation has been developed that includes boxes or blocks to represent activities or functions and various connecting arrows that have specific representational purposes.

The IDEF3 model is similar to a decision tree with information boxes associated to each decision point. The goal is to concentrate on the structuring of the specific thought processes followed by NC programmers. In this way, the information necessary to build a decision processor can be stored independently of the AI technique utilized. This research utilized graphical models and a database for holding machining strategies acquired from literature and interviews.

Product Models. The complete electronic product definition for a manufactured part consists of all the information related to designing, producing, inspecting, and managing the manufacture of that part. Efficient utilization of these articles of data would facilitate the rapid production of the item using computerized machine tools and processes. National efforts are underway to define standards for complete product models. A product model is defined as a computer representation of a part or assembly containing the combined information associated with CAD, CAM, CAE, CAI, process planning and job tracking. It is the basis for Computer Integrated Manufacturing (CIM), Concurrent Engineering (CE), and Design For Manufacturability (DFM) studies. Figure 1 identifies the major areas and critical components

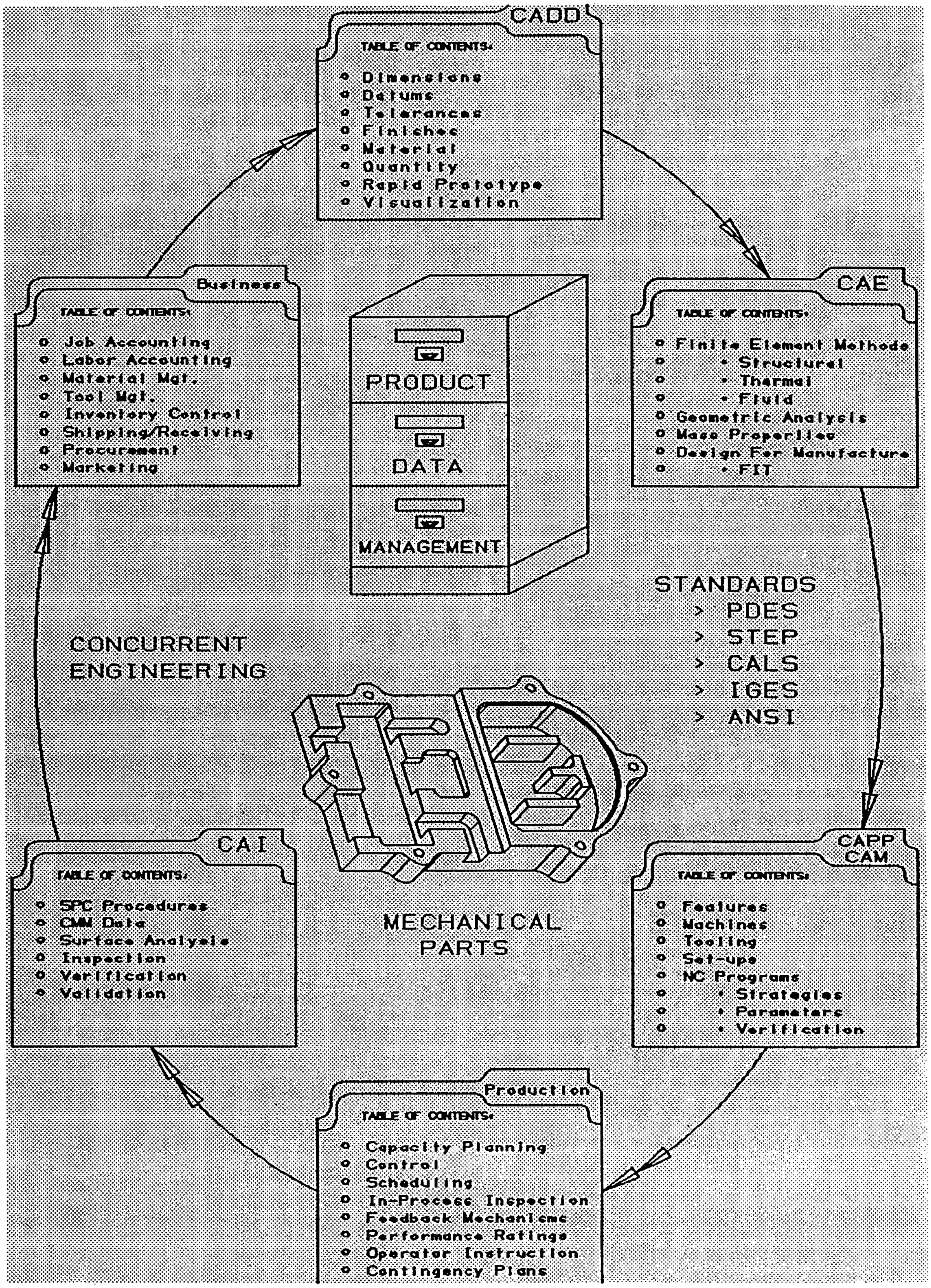


Figure 1: Components of the Complete Product Model

within each area necessary to build a complete product model for a mechanical part. The file folders represent product data that should be stored in a common area so that it can be shared throughout the operation.

The data provided by the product model must be accessible, well organized, and conform to standards prevalent in both industry and government. In many ways, the product model is unobtainable because it relies on the unrestricted sharing of data throughout the design, engineering, and manufacturing processes. Many aspects of converting raw material into a finished parts are difficult to track on the shop floor. The Computer Aided Acquisition and Logistics Support (CALS) and Product Definition Exchange Specification (PDES) / Standard for the Exchange of Product Data (STEP) initiatives are currently addressing these very issues.

MACHINING STRATEGY

"The first principle of knowledge engineering is that the problem solving power exhibited by an intelligent agent's performance is primarily the consequence of its knowledge base, and only secondarily a consequence of the inference method employed. Expert systems must be knowledge-rich even if they are methods-poor. This is an important result and one that has only recently become well understood in AI. For a long time AI has focused its attentions almost exclusively on the development of clever inference methods; almost any inference method will do. The power resides in the knowledge."

-Edward Feigenbaum, Stanford University

Available tooling, machine accuracy, and setup rigidity can influence one's decision drastically. Many times, duplicate tools are used to both rough and finish a groove (especially when considering a large number of parts) so that results can be repeated consistently. Setup decisions are based upon the configuration of the part, the material and the tooling available. Setup choices include tool selection, depth/width/cut, speed and feed.

Features. The concept of "features" plays an important role in linking design to manufacturing. CAM-I's Current Status of Features Technology defines features used in process planning for manufacturing as "shapes and technological attributes associated with manufacturing operations and tools [3]." From a machinist's viewpoint, the first step in developing a machining plan is to break the part into features. Each feature is characterized as a structural entity whose attributes specify lower-level geometry, topology, tolerance, surface finishes, etc.

Tool Selection. Roughing and Finishing tools are capable of cutting in a wide variety of materials and orientations. For clearance purposes, the angles of the tool and it's reference to the work and holder are critical. Although many tools are custom ground, standardized tooling inserts are becoming common and general tool definitions can be identified for classes of materials and conditions. Clearance for chip removal, part avoidance, and holders is one of the first considerations. Finish cuts for various materials are usually between 0.010" and .075" depending on the material and desired finishes and tolerances. "Spring" or "free" passes are also common. No new material is theoretically cut but, due to part and tool deflection, this pass usually cuts 0.0005"-.001" and leaves a smooth finish.

Speeds and Feedrates. The subject of speeds and feeds could never be generalized for all applications due to the wide variety of materials, machines, and tools available. The usual sequence is to start with a cutting speed (surface feet per minute) for the particular work material, calculate an RPM using the speed and diameter of the part, then use the RPM and feed per tooth for the tool/material to calculate a feedrate and are usually assigned for approach, intermediate, and exit motions. As a safety precaution, the feedrate can be used to calculate the material removal rate and the required horsepower. Speeds and feeds are commonly adjusted once the operation begins. Sounds, torque readings, and vibration are key signals which can be used to tune cutting parameters during machining.

Tool Motion Points. In any CAM system, many tool location points must be defined. The first considerations are for safety and non-violation of the finished part boundaries. Home or from points are defined either uniquely for a machine or arbitrarily by the operator so as to reference the tool path correctly before beginning the operation. It is commonly some fixed distance from the part origin (usually at the end of the part on the axis of rotation). Approach points are used as an place where the tool can rapid feed to before entering the material. The tool enters the material to the first depth of cut and stops at the entry point. Motions are executed to complete one pass before the tool moves to the exit point. The exit point to the withdrawal point can either be an intermediate move before moving back to re-enter the part or a place from which the tool can rapid feed back to the home location. In addition there are assignable intermediate approach and re-entry points as are required for profiles with multiple passes. Much of tool wear occurs when the tool violates un-represented stock on entry and exit motions. Strategic selection of these points, along with the feedrates assigned at each stage, can optimize and ensure the safety of tool paths.

Tool Selection. Often times the same tool is used for both roughing and finishing grooves. In production situations, the concept of duplicate tooling is employed to keep tooling costs down and achieve repeatability. The tool selection process can be separated into a series of choices between materials and geometric attributes associated with grooving tools. The choices include a tool material, tool width, tool shape, tool length, and corner radius.

Cut Width and Depth. The width of cut is directly based on the material's machinability and the tool selected. The first, and very critical cut is at full tool width. A very tough material may be cut at somewhere between 1/4 and 1/2 of it's width on passes following the plunge cut. Grooves are cut, most often, to full depth on every pass. It is different than say, milling, because tool pressures are close to the same at full depth as when first entering the material. If a multi-depth groove is encountered, it is often treated as two operations, with a smaller tool repeating the same basic sequence as the larger tool.

Approach and Entry. The tool should start at the home position (may be fixed or reset at some fixed distance from the origin of the part) and move at rapid feed over to .020" away from the feature. The wall closest to either the chuck or the center should be used at the start location in turning. It is safe to move the tool at a rapid feedrate over to within .100" outside the stock outline or above the part. The firstlmotion can be a direct plunge, at full tool width, to a depth prescribed or a ramp. A dwell should be programmed at about one second to allow for chip removal.

KNOWLEDGE ACQUISITION AND REPRESENTATION

A machinist in charge of designing a tailored process plan is always faced with the problem of selecting from a vast variety of possible combination values of machining parameters, such as speed, feed, depth of cut, tool material and size, tool entry strategy, cutting strategy,...etc. A machinist makes decisions based on past machining cases that he recalls from his experience. Very often, to help overcome some difficulties of this unstructured decision-making process, machinists employ some form of heuristic rules to distinguish between different cases; however, each machining case can be unique in some aspects and may require task-specific solution strategies. "It is very difficult to construct general, and yet simple, rules capable of handling all or most machining cases," said Joe Alejandrez, a 20-year-experience machinist form the Physical Science Laboratory (PSL). That is because process planning is a function of multiple inputs and multiple outputs describing the part to be machined, the manufacturing resources, and the manufacturing procedures. Each output decision is driven by a set of raw interacting inputs and possibly by other outputs, resulting in a complex parallel/sequential decision process. The process is subject to machinists' preference as there is a wide range of machining parameters that can satisfy the requirements of the product model. This wide range is the basis of much discussion regarding the 'best' approach to machining a quality part. A trained machinist can effectively recognize and evaluate all pertaining factors and work out a 'good' process plan.

The most important and difficult aspect of this research is to develop effective tools with which to elicit knowledge from domain experts. The skills obtained through years of observation training, improvising, and trial-and-error are not easy to explain verbally, document, or formalize for the purpose of building automated NC programming systems. A significant amount of time must be spent in familiarizing oneself with the field and terminology before attempting to solicit strategies directly from the NC programmers and machinists. Even though technology has matured very quickly in the areas of CNC machining and CAD/CAM, experience with manual machining, "old tricks," and traditional material removal skills are still necessary. This practical, applied way of thinking cannot be documented in machining handbooks or obtained from questionnaires and interviews. An aged work-force, loss of manufacturing jobs, and the disappearance of apprenticeship and mentor programs combine to put the U.S. in danger of losing a once dominant technology.

Machine operators and programmers must work as a team in order for high standards of quality to be achieved. Many times, process and design engineers have not been included in this team. NC programming functions are rarely evaluated from an engineering standpoint. Many times either technicians and craftspeople or management decide on the tools that will be made available for process planning. To keep production going, the system a company is familiar with becomes a safety net. New options are rarely given a chance to succeed due to the steep learning curve and high costs associated with CAD/CAM, NC programming, and post-processing.

It is difficult to obtain the assistance of skilled machinists when attempting to build an automated programming system. Information is difficult to communicate verbally and formalize for the purpose of designing a knowledge base. A key factor in the effort lies in the re-training of shop-floor workers. If some decisions they normally make are being replaced by computer programs, then new knowledge must be given in return. The services of an experienced lathe operator (Ken Davis, 17 years between FreightLiner and BOEING) for two months, greatly contributed to the success of the this project.

This research helped to develop knowledge elicitation and representation tools that can be used to formalize the expertise gained through machining strategy meetings, literature review, and programmer feedback. It includes the design of questionnaires that can be used to document strategies in a set format during interviews and observation training.

Developing knowledge acquisition and representation tools which allow information pertaining to specific turned features to be documented and evaluated is viewed as an important part of a production-worthy CAPP system. Generative AI programming languages all require the same knowledge as the basis for structuring a decision processor, regardless of the technique. Many CAPP systems require AI programmers to modify and enhance the rule bases. Breaking the decision process down into small steps allows for a procedural approach to the actual decision processing. Knowledge attained during meetings with experts along with the resulting rules formulated for that particular decision can be stored in the model. (See **Process Flow Model)**

Automatic creation of NC programs requires; 1) user-interaction -- to gather location specific operating information, 2) interrogation of the features geometric description within the CAD database, 3) a file based communication link to a decision processor and, 4) An applications interface language capable of building tool paths from user generated programs. Very few commercial CAD/CAM systems provide the user with enough power to accomplish these tasks. Many CAPP systems have had to build their own primitive CAD system for input or used symbolic languages to output the parts geometric description. Obvious extensions to the automated process planner described here include automatic feature recognition, enhanced decision processing methods, part program optimization routines, and Group Technology (GT) coding schemes for individual rotational features. For this project we utilized ANVIL5000 CADD/CAM/CAE software for its application interface and integrated capabilities.

ARTIFICIAL NEURAL NETWORKS

Interest in 'brain style computing' in terms of artificial neural networks (ANN) is increasing. ANNs are parallel distributed processing architectures composed of a large number of small interacting elements that are massively interconnected. Each of these elements sends excitatory and inhibitory signals to other units that in turn update their behaviors based on these received messages. ANNs emulate the functionality of the human brain that implicitly stores knowledge in the interconnection weights, and not in the neurons themselves [9]. A typical network consists of a set of processing elements (PE) grouped hierarchically in layers and interconnected in some fashion. These PEs sum *N* weighted inputs and transfer that outcome through a mathematical function. One of the most popular neural paradigms is the multilayer perceptron trained using the Error Back Propagation Training Algorithm (EBPTA) [6].

Neural network models have key advantages over rule-based systems. First, the learning process, or the knowledge acquisition process, takes place by presenting only sample machining cases to the network. Learning occurs through modifying the connection weights between network elements (by a learning algorithm) to model the representation of the cases [10]. Knowledge is thus stored implicitly in the final interconnections values. Second, because training examples need not be exhaustive, it is conceivable that neural network methodology could ease the knowledge acquisition bottleneck that is hampering the creation of rule-based systems [11]. Finally, when utilized as a decision processor, an artificial neural has inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization.

A NEW KNOWLEDGE ACQUISITION METHODOLOGY

The adopted research methodology is aimed at investigating the utilization of a neural-based knowledge acquisition approach. A neural-based process planning system is a knowledge-based system, where knowledge is stored implicitly in the network parameters during the course of learning. As described earlier, learning takes place gradually when segmented pieces of knowledge (i.e. training examples) are presented to the network and accumulated in its memory. Based on this operational mode of neural systems, and taking advantage of our knowledge and experience in developing rule-based systems for similar tasks (i.e. milling assistant [3]). This methodology should satisfy a number of basic requirements. First, it must reduce the burden associated with developing rule-based systems by avoiding constructing general rules, because, as explained earlier, experts are not good at generalizing solutions. Second, this methodology must ensure that knowledge be cast in a form usable by neural systems; forms of rules are excluded. What is required by a neural system is a set of training examples, where each example represents a real machining case described by a vector of numeric values of process variables encoded in a convenient from. Third, consequently, the new methodology must employ effective instrumental knowledge gathering forms; and since no general inferential statements are required at this stage, the knowledge gathering forms can be mere data collection forms. The basic idea underlying this procedure is to enable the user provide relevant training examples only, and the neural network, through the learning algorithm, will implicitly derive the IF--THEN rules.

First, a neural network paradigm must be selected in consensus with the application at hand. In this research, a *supervised* neural network model is employed as a *classifier*. A supervised learning model is selected as it enables the expert to express his judgmental opinion in a relative manner. This feature is very essential because, in this application, it is not required to group the training examples into clusters as would be the outcome if an *unsupervised* learning model were used. Instead, every single training example be explicitly identified (by the expert) as belonging to a specific class that represents a machining decision. The neural model is employed as a classifier so that the output decision belongs to a class of objects (e.g. type tool material: HSS, Carbide, Ceramic), or to a ranges of numeric values (e.g. tool size: 1/4"-1/2", 1/2"-1",...). This method helps considering the variety of machining choices that can satisfy the requirements of a quality product.

The second decision is indeed part of the knowledge acquisition procedure. It is concerned with selecting appropriate process planning decisions parameters that can be adequately handled by the described neural model. The parameters chosen for the purpose of this study served as real illustrative examples of the new methodology implementation. These functions are chosen to illustrate our methodology because the output of each function belongs to one of a finite number of well-defined output classes that are generally encountered in practice.

Having decided on the output PP decision variables to be automated, the next step was to find all related PP input variables that could affect the selection of the output variable. This process took place in an iterative manner as follows:

Initially, a long list of possible input variables was constructed that could collectively drive the selection process of all output variables simultaneously.

It was found that not all selected input variables do contribute to the selection of a given output decision value. Thus, to avoid confusion among various inputs and outputs, each output decision variable and its inputs were filtered out and handled separately. Four individual forms were then designed, one each for output decision.

Working with individual forms, the meanings of variables needed further explanation. This was an important step as it a created jargon of well-defined terms that were conveniently used by experts. This revisited understanding revealed that some of the input variables had to be deleted as they did not affect selecting the value of the output parameter. At this point, our forms were judged to provide 'good' instruments to start collecting data. Each form deals with one output variables, and only relevant non-redundant input variables.

During data collection, our expert machinists tended to emphasize some input variables in certain cases, while giving less weight to the same variables when dealing with other cases. To take this distinction into account, a new column was added to all data collection forms to identify the weight or the importance of an input variable at a particular instant. Weights were assigned to variables according to 0-10 scale in an increasing order of merit.

Using the data forms, a good number of examples were constructed (about 50) for output decision. At this stage, it was observed that some input variables constantly scored low weight rating (e.g. 2, 3). Those variables were deleted form the forms as they were considered to represent noise. This also reduced the dimensionality of the input vector.

In light of the above requirements, a neural-based machining knowledge acquisition technique is designed and implemented during the course of developing a neural model as a supportive decision making tool in machining process planning.

CONCLUSIONS

Computerized machining processes are slowly becoming commonplace in both large and small companies. Current research into knowledge based process planning has gained limited acceptance in shops because of the complexity and shortcomings of such systems. One problem is that the artificial intelligence technique often becomes the focus of academic research projects. The complexity of AI and computer science can severely limit a systems' applicability in production environments. Machine operators and programmers must be included in the design and development of such applications. Many systems are dependent on a skilled computer programmer to tune the knowledge and rules embedded in, what is often, very cryptic code. The transfer of knowledge between the process engineer and the programmer can be inefficient and frustrating. Creative inference techniques and clever knowledge structures cannot solve problems that are not well understood by the applications' designers.

The tradeoffs between CAD/CAM and traditional NC programming languages (i.e., MDI, APT, etc.) remains a heated topic among manufacturers. The inability of software to reliably control production processes has led many to always have a safety net of "the old way." The focus is often taken off the overall process and put on the machinist who is planning, verifying, and executing the programs.

ACKNOWLEDGEMENTS

The project is grateful to Army Research Office, Boeing Corp., Ladish Corp., Manufacturing and Consulting Services, Physical Sciences Laboratory and Sandia National Laboratories for their contributions of grants, expertise, equipment and software.

REFERENCES

[1] Anderson, D.C., and Chang, T.C., "Geometric Reasoning in Feature-Based Design and Process Planning," *Comput. & Graphics*, Vol. 14, No. 2, pp. 225-235, 1990.

[2] D. Bedworth, R. Henderon, M. Wolfe, *'Computer-Integrated Design and Manufacturing'*,McGraw-Hill, 1991.

[3] Burd, W.C., *The Milling Assistant,* SAND89-1519, Computer Aided Manufacturing Division, 7483, Sandia National Laboratories, Albuquerque, New Mexico, 1989.

[4] T. Chang, R. Wysk, H. Wang, '*Computer-Aided Manufacturing*', Prentice Hall, NJ, 1991.

[5] Chung, J.C.H., Patel, D.R., and Cook, R.L., "Feature-Based Modeling for Mechanical Design," *Comput. & Graphics*, Vol. 14, No. 2, pp. 189-199, 1990.

[6] Dayhoff J.E., '*Neural Networks Architectures: An Introduction*', van Nostrand International Company, New York, 1990.

[7] Joshi, Vissa, and Chang, "Expert Process Planning System with a Solid Model Interface," Chapter 6, *Knowledge-Based Systems in Manufacturing*, Taylor and Francis Inc. Philadelphia, PA, 1989.

[8] Phillips, R.H., and Mouleeswaran, C.B., *A Knowledge-Based Approach to Generative Process Planning*, presented at the CASA/SME Autofact 85 Conference, November 1985.

[9] Rumelhart D.E., McClelland J.L., and the PDP Research Group, '*Parallel Distributed Processing: Explorations in the Microstructure Cognition*', Volume 1, MIT Press, 1986.

[10] Shalkof R.J., '*Pattern Recognition: Statistical, Structural, and Neural Approaches*', John Wiley & Sons, Inc., USA, 1992.

[11] Zurada J.M., '*Introduction to Artificial Neural Systems*', West Publishing Company, USA, 1992.

TASK 3.1

PARTICIPATING SCIENTIFIC PERSONNEL

TASK 3.1 PARTICIPATING SCIENTIFIC PERSONNEL

NAME	STATUS
C. Cameron Bumgardner	BS Candidate
Prof. L. D. Cox	Associate Professor
Dr. D. E. Culler	PhD Awarded
Dr. A. Al-Ghanim	PhD Awarded
John D. Holder	BS Awarded
Satish Kalaparthi	MS Awarded
Robert I. McElroy	PhD Candidate
Dr. R. J. Pederson	Associate Professor

ADAPTIVE CONTROL OF MANUFACTURING PROCESSES

Richard Colbaugh
Kristin Glass

Department of Mechanical Engineering
New Mexico State University

Final Report for Task 3.2 of Applied Manufacturing Research
Submitted to the Army Research Office

February 1996

SUMMARY

This report documents the work performed on the project "Adaptive Control of Manufacturing Processes" for the Army Research Office. The primary technical objectives of the project were as follows:

1.) Develop, implement, and experimentally verify adaptive controllers for unconstrained and constrained robotic systems.

2.) Develop, implement, and experimentally verify adaptive estimation and control algorithms (including neural networks) for other important manufacturing systems. Problems of interest include the real-time estimation of machine tool wear using cutting-force measurements, the identification and compensation of thermal effects in machining, and the control of cutting force in machining.

The report is composed of reprints of ten recent and representative papers which describe aspects of the research. A full list of the project publications is given below; those included in the report are marked with an asterisk. Additional information regarding these papers or the project will be provided by the investigators upon request.

PROJECT PUBLICATIONS

1. Journal Articles and Book Chapters

* Colbaugh, R. and K. Glass, "Decentralized Adaptive Control of Nonholonomic Mechanical Systems" (submitted for publication)

Colbaugh, R. and E. Barany, "Adaptive Control of Nonholonomic Robotic Systems" (submitted for publication)

Colbaugh, R. and K. Glass, "Adaptive Task-Space Control of Flexible-Joint Manipulators" (submitted for publication)

* Colbaugh, R. and K. Glass, "Adaptive Compliant Motion Control of Electrically-Driven Manipulators" (submitted for publication)

Colbaugh, R. and K. Glass, "Adaptive Compliant Motion Control of Manipulators

Without Velocity Measurements" (submitted for publication)

Colbaugh, R. and K. Glass, "Adaptive Tracking Control of Manipulators Using Only Position Measurements" (submitted for publication)

Wedeward, K. and R. Colbaugh, "Adaptive Controllers Which Employ Saturated Error Update Laws for Robot Trajectory Tracking" (submitted for publication)

* Colbaugh, R. and K. Glass, "Adaptive Stabilization of Mechanical Systems Using Only Configuration Measurements" (submitted for publication)

Barany, E. and R. Colbaugh, "Adaptive Stabilization of Mechanical Systems" (submitted for publication)

Barany, E. and R. Colbaugh, "Adaptive Control of Chaotic Dynamical Systems" (submitted for publication)

* Flachs, S. and R. Colbaugh, "Neural Network Control of Thermal Expansion" (submitted for publication)

* Colbaugh, R., K. Glass, and E. Barany, "Adaptive Regulation of Manipulators Using Only Position Measurements", *International Journal of Robotics Research*, Vol. 15, 1996 (in press)

Seraji, H. and R. Colbaugh, "Force Tracking in Impedance Control", *International Journal of Robotics Research*, Vol. 15, 1996 (in press)

Colbaugh, R. and K. Glass, "Decentralized Adaptive Control of Electrically-Driven Manipulators", *Journal of Intelligent Control and Systems*, Vol. 10, 1996 (in press)

Colbaugh, R., E. Barany, and K. Glass, "Adaptive Control of Constrained Robotic Systems for Waste Management Applications", *International Journal of Environmentally Conscious Design and Manufacturing*, Vol. 5, 1996 (in press)

Colbaugh, R., K. Wedeward, and A. Engelmann, "Adaptive Compliant Motion Control of Manipulators: Theory and Experiments", *Journal of Robotic Systems*, Vol. 13, 1996 (in press)

* Colbaugh, R. and K. Glass, "A Robust Approach to Real-Time Tool Wear Estimation", *International Journal of Robotics and Automation*, Vol. 11, No. 1, 1996 (in press)

Seraji, H., R. Colbaugh, and K. Glass, “Control of a Serpentine Robot for Inspection Tasks”, *NASA Tech Briefs Journal*, Vol. 20, 1996 (in press)

Colbaugh, R., K. Glass, and E. Barany, “A Mechatronic Systems Approach to Controlling Robotic Systems With Actuator Dynamics”, invited chapter in *Mechatronic System Techniques and Applications*, Gordon and Breach International Series in Engineering, Technology, and Applied Science, 1996

Colbaugh, R., K. Glass, and E. Barany, “Adaptive Stabilization and Tracking Control of Electrically-Driven Manipulators”, *Journal of Robotic Systems*, Vol. 13, No. 4, 1996

Colbaugh, R., K. Wedeward, K. Glass, and H. Seraji, “New Results on Adaptive Impedance Control for Dexterous Manipulators”, *International Journal of Robotics and Automation*, Vol. 11, No. 1, 1996

Colbaugh, R., K. Glass, and H. Seraji, “Performance-Based Adaptive Tracking Control of Robot Manipulators”, *Journal of Robotic Systems*, Vol. 12, No. 8, 1995

* Colbaugh, R., H. Seraji, and K. Glass, “Adaptive Compliant Motion Control for Dexterous Manipulators”, *International Journal of Robotics Research*, Vol. 14, No. 3, 1995

* Glass, K., R. Colbaugh, D. Lim, and H. Seraji, “Real-Time Collision Avoidance for Redundant Manipulators”, *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, 1995

Colbaugh, R. and K. Glass, “Decentralized Adaptive Control of Mechanical Systems”, *Intelligent Automation and Soft Computing: An International Journal*, Vol. 1, No. 3, 1995

Colbaugh, R. and K. Glass, “Robust Adaptive Control of Redundant Manipulators”, *Journal of Intelligent and Robotic Systems*, Vol. 14, No. 1, 1995

Colbaugh, R. and K. Glass, “Decentralized Adaptive Compliance Control of Robot Manipulators”, *Robotica*, Vol. 13, No. 5, 1995

* Colbaugh, R., “Stability Analysis for a Class of Neural Networks”, *International Journal of Robotics and Automation*, Vol. 10, Vol. 4, 1995

Colbaugh, R., K. Glass, and H. Seraji, “Robust Adaptive Control of Lagrangian Systems", *Computers and Electrical Engineering: An International Journal*, Vol. 21, No. 4, 1995

Colbaugh, R., H. Seraji, and K. Glass, "A New Class of Adaptive Controllers for Robot Trajectory Tracking", *Journal of Robotic Systems*, Vol. 11, No. 8, 1994

Colbaugh, R., H. Seraji, and K. Glass, "Decentralized Adaptive Control of Manipulators", *Journal of Robotic Systems*, Vol. 11, No. 5, 1994

Colbaugh, R., K. Glass, and M. Jamshidi, "Advanced Controller Development for Robotic Waste Management", *International Journal of Environmentally Conscious Design and Manufacturing*, Vol. 3, No. 3-4, 1994

Colbaugh, R., "Adaptive Point-to-Point Motion Control of Manipulators", *International Journal of Robotics and Automation*, Vol. 9, No. 2, 1994

Colbaugh, R., K. Glass, and P. Pittman, "Adaptive Control for a Class of Hamiltonian Systems", *Computers and Electrical Engineering: An International Journal*, Vol. 20, No. 1, 1994

Colbaugh, R., K. Glass and H. Seraji, "Direct Adaptive Control of Robotic Systems", *Journal of Intelligent and Robotic Systems*, Vol. 9, No. 1, 1994

Colbaugh, R. and K. Glass, "Adaptive Control of Manipulators Handling Hazardous Waste", *Radioactive Waste Management and Environmental Restoration*, Vol. 18, No. 1, 1994

Colbaugh, R., J. Baca, and M. Jamshidi, "Decentralized Adaptive Compliance Control of Manipulators for Waste Management Applications", Chapter 6, *Robotics and Remote Systems in Hazardous Environments*, Prentice Hall, 1994

Akbarzadeh, M., K. Kumbla, M. Jamshidi, and R. Colbaugh, "Intelligent Control of Flexible and Redundant Robots", Chapter 5, *Robotics and Remote Systems in Hazardous Environments*, Prentice Hall, 1994

2. Refereed and Invited Conference Publications

Colbaugh, R. and K. Glass, "Decentralized Adaptive Stabilization and Tracking Control of Electrically-Driven Manipulators", *Proc. 1996 IFAC Triennial World Congress*, San Francisco, CA, July 1996

Glass, K. and R. Colbaugh, "Sensor-Based Coordinated Motion Control of Redundant Manipulators", *Proc. Sixth International Symposium on Robotics and Manufacturing*, Montpellier, France, May 1996 (invited paper)

Colbaugh, R. and K. Glass, "Robust Adaptive Regulation of Manipulators: Theory and Experiments", *Proc. 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996

Colbaugh, R. and K. Glass, "Adaptive Compliant Motion Control of Manipulators Without Velocity Measurements", *Proc. 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996

* Glass, K. and R. Colbaugh, "Real-Time Tool Wear Estimation Using Cutting Force Measurements", *Proc. 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996

Colbaugh, R. and K. Glass, "Adaptive Tracking Control of Manipulators Using Only Position Measurements", *Proc. 34th IEEE Conference on Decision and Control*, New Orleans, LA, December 1995

Wedeward, K. and R. Colbaugh, "Adaptive Controllers Which Employ Saturated Error Update Laws for Robot Trajectory Tracking", *Proc. 34th IEEE Conference on Decision and Control*, New Orleans, LA, December 1995

Colbaugh, R. and K. Glass, "Adaptive Control of Electrically-Driven Manipulators: Theory and Experiments", *Proc. 1995 IEEE Conference on Control Applications*, Albany, NY, September 1995

Wedeward, K. and R. Colbaugh, "Adaptive Position/Force Regulation Without Rate Information", *Proc. 1995 IEEE Conference on Control Applications*, Albany, NY, September 1995

Colbaugh, R. and K. Glass, "Adaptive Tracking Control of Rigid-Link Electrically-Driven Manipulators", *Proc. 10th IEEE International Symposium on Intelligent Control*, Monterey, CA, August 1995

Colbaugh, R. and K. Glass, "Real-Time Tool Wear Estimation Using Recurrent Neural Networks", *Proc. 10th IEEE International Symposium on Intelligent Control*, Monterey, CA, August 1995

Colbaugh, R. and E. Barany, "Stability Analysis for a Class of Neural Networks", *Proc. 10th IEEE International Symposium on Intelligent Control*, Monterey, CA, August 1995

Colbaugh, R., K. Wedeward, K. Glass, and H. Seraji, "New Results on Adaptive Impedance Control for Dexterous Manipulators", *Proc. 10th IEEE International Symposium on Intelligent Control*, Monterey, CA, August 1995

Colbaugh, R. and K. Glass, "Adaptive Regulation of Rigid-Link Electrically-Driven Manipulators", *Proc. 1995 IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 1995

Colbaugh, R., K. Glass, and E. Barany, "Adaptive Output Stabilization for a Class of Mechanical Systems", *Proc. SIAM Conference on Control and Its Applications*, St. Louis, MO, April 1995 (winner of Best Presentation Award for the conference)

Barany, E. and R. Colbaugh, "Adaptive Control and Nonlinear Dynamics", *Proc. SIAM Conference on Control and Its Applications*, St. Louis, MO, April 1995

Colbaugh, R., K. Glass, and E. Barany, "Adaptive Output Stabilization of Manipulators", *Proc. 33rd IEEE Conference on Decision and Control*, Orlando, FL, December 1994

Wedeward, K. and R. Colbaugh, "Adaptive Compliant Motion Control of Position-Controlled Manipulators", *Proc. 33rd IEEE Conference on Decision and Control*, Orlando, FL, December 1994

Colbaugh, R., K. Glass, and H. Seraji, "Decentralized Adaptive Tracking Control of Robot Manipulators", *Proc. IFAC Symposium on Robot Control*, Capri, Italy, September 1994

Colbaugh, R., K. Glass, and A. Engelmann, "Decentralized Adaptive Compliance Control of Robot Manipulators", *Proc. IFAC Symposium on Robot Control*, Capri, Italy, September 1994

Colbaugh, R., K. Glass, and H. Seraji, "Adaptive Compliance Control for Redundant Manipulators", *Proc. 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Munich, Germany, September 1994

Colbaugh, R. and K. Glass, "Decentralized Adaptive Control of Mechanical Systems", *Proc. Workshop on Robust Control via Variable Structure and Lyapunov Techniques*, Benevento, Italy, September 1994

Colbaugh, R., K. Glass, and H. Seraji, "Application of Adaptive Tracking Control to Industrial Manipulators", *Proc. 1994 IEEE Conference on Control Applications*, Glasgow, Scotland, U.K., August 1994

Colbaugh, R., K. Glass, and H. Seraji, "Control of a Serpentine Robot for Inspection Tasks", *Proc. Fifth International Symposium on Robotics and Manufacturing*, Maui, HA, August 1994 (invited paper)

Colbaugh, R., K. Wedeward, and A. Engelmann, "Adaptive Compliant Motion Control of Redundant Manipulators: Theory and Experiments", *Proc. Fifth International Symposium on Robotics and Manufacturing*, Maui, HA, August 1994 (invited paper)

Colbaugh, R., K. Glass, and H. Seraji, "Adaptive Compliance Control of Manipulators", *Proc. 1994 American Control Conference*, Baltimore, MD, June 1994

Colbaugh, R. and H. Seraji, "Adaptive Tracking Control of Manipulators: Theory and Experiments", *Proc. 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994 (nominated for Best Conference Paper Award)

Colbaugh, R. and A. Engelmann, "Adaptive Compliant Motion Control of Manipulators: Theory and Experiments", *Proc. 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994

Colbaugh, R., "Topological and Geometrical Aspects of Robotics" *New Mexico Geometry and Topology Seminar*, Las Cruces, NM, February 1994 (invited paper)

TASK 3.2

PUBLICATIONS AND TECHNICAL REPORTS PUBLISHED

submitted for publication

DECENTRALIZED ADAPTIVE CONTROL OF NONHOLONOMIC MECHANICAL SYSTEMS

R. Colbaugh **K. Glass**

Department of Mechanical Engineering
New Mexico State University, Las Cruces, NM 88003 USA

Abstract

This paper considers the problem of controlling nonholonomic mechanical systems in the presence of incomplete information concerning the system model and state, and presents a class of decentralized adaptive controllers as a solution to this problem. The proposed control strategies provide simple and robust solutions to a number of important nonholonomic system control problems, including stabilization to an equilibrium manifold, stabilization to an equilibrium point, and trajectory tracking control. All of the schemes are computationally efficient, are implementable without system dynamic model or rate information, and ensure uniform boundedness of all signals and accurate motion control. Computer simulation results are provided to complement the theoretical developments.

1. Introduction

There is great theoretical and practical interest in controlling mechanical systems in the presence of measurement and model uncertainty. It is well-known that control schemes which require precise knowledge of the complete system model to provide good performance suffer from many serious limitations. Additionally, it is usually unrealistic to assume that the entire system state can be accurately measured and used for feedback; indeed, although high-precision sensors are available for measuring the configuration of mechanical systems, rate measurements are typically either contaminated with noise or not available at all. Recognition of these facts has motivated considerable interest in developing control methods which can be successfully implemented despite model and state uncertainties. This challenging problem becomes even more difficult in the important case in which the system is subjected to holonomic or nonholonomic (nonintegrable) constraints on its kinematics.

Nonholonomically constrained mechanical systems, in particular, have attracted significant attention recently. Much of this interest is a consequence of the importance of such systems in applications. For example, nonholonomic constraints arise in systems with rolling contact, such as wheeled vehicles and multifingered robotic hands, and in systems for which the dynamics admits a symmetry, such as space structures with angular momentum conservation; observe that a wide variety of systems of practical importance are constrained in this way. Interest in studying nonholonomic systems is also motivated by their role as a class of "strongly" nonlinear systems for which traditional nonlinear control methods are insufficient and new approaches must be developed. For instance, the investigation of these systems has led to recent progress in geometric control theory and geometric mechanics.

Most of the work reported to date on controlling nonholonomic mechanical systems has focused on the steering or trajectory generation problem; the interested reader is referred to [1,2] for a survey of some of this work and to [3-9] for an indication of recent progress. This research has produced numerous interesting solutions to the trajectory generation problem, and has led to advances in nonlinear control theory and practice. However, the work has proceeded by considering the system velocities to be the control inputs and working with the system kinematic model, thereby addressing the problem at the *kinematic control* level and ignoring the mechanical system dynamics.

There are important reasons for formulating the nonholonomic system control problem at the *dynamic control* level, where the control inputs are those produced by the system actuators and the system model contains the mechanical system dynamics. For example, since this is the level at which control actually takes place in practice, designing controllers at this level can lead to improved implementability and can help in the early identification and resolution of difficulties. It is interesting to consider the problem of controlling holonomically constrained robots in this regard: the kinematic control problem in this case is trivial, but the dynamic control problem is often quite challenging [e.g., 10]. Another motivation for considering the dynamic control problem is that many nonholonomic systems are most naturally studied at this level, so that such an approach broadens the scope of potential applications [e.g., 11]. Recognizing the importance of addressing the nonholonomic system control problem at the dynamic control level, some researchers have considered this problem in recent years [e.g., 2, 12-17]. Significant progress has been made in understanding the fundamental characteristics of these systems, and several useful dynamic controllers have been proposed. However, virtually all of the dynamic control strategies proposed to date have been developed by assuming that the full dynamic model is precisely known and that the entire system state is measurable.

This paper considers the dynamic control of uncertain nonholonomic mechanical systems, and proposes that the problem be studied within an adaptive control framework. The majority of the research in adaptive control of mechanical systems has proceeded by assuming that the uncertainty in the system model can be linearly parameterized using constant coefficients [e.g., 18-21]. This assumption permits the design of controllers for which system stability can be ensured, but it can be limiting. For example, controllers developed using this approach are typically very complex, require full state measurement, are not easily implementable in terms of arbitrary generalized coordinates, and have exhibited poor robustness and transient performance. These characteristics suggest that this methodology may not be well suited for applications involving nonholonomic systems. Furthermore, this formulation leads inevitably to centralized algorithms, in which the control input for each configuration degree-of-freedom (DOF) depends on all of the other DOF; this is a consequence of the need to incorporate the system dynamic model directly into the control law. In contrast with this approach to adaptive robot control, the present schemes are developed using the recently proposed *performance-based adaptive control* methodology, in which the adaptive laws adjust the controller gains based on system performance rather than on knowledge of the manipulator model [22-25]. As a consequence, it is possible to design the schemes so that they are computationally efficient, ensure good robustness and transient performance, and are implementable in terms of any set of generalized coordinates without rate measurements or information regarding the system dynamic model. Additionally, the controllers can be given a simple decentralized structure, in which the control input for each configuration DOF is computed based on information concerning only that DOF. Note that decentralized control offers significant potential benefits relative to centralized control, including enhanced computational simplicity, ease of implementation, increased robustness and fault tolerance, and reduced reliance on system model information. The control strategies proposed in this paper provide solutions to a number of important nonholonomic system stabilization and trajectory tracking problems, and all are shown to ensure uniform boundedness of all signals and accurate motion control. The efficacy of the proposed approach is illustrated through computer simulations with two nonholonomic mechanical systems: a mobile robot with rolling constraints and a planar mechanism for which angular momentum is conserved.

The paper is organized in the following manner. Some preliminary facts of relevance to the nonholonomic system control problem are established in Section 2. In Sections 3 through 5 the decentralized adaptive stabilization and tracking schemes are presented and analyzed and the performance of the proposed controllers is illustrated through computer simulation. Finally, Section 6 summarizes the paper and provides some suggestions for future work.

2. Preliminaries

Consider the class of nonholonomic mechanical systems described by the equations [e.g., 12]

$$M(\mathbf{x})\mathbf{T} = H(\mathbf{x})\ddot{\mathbf{x}} + V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}(\mathbf{x}) + A^T(\mathbf{x})\lambda \tag{1a}$$

$$A(\mathbf{x})\dot{\mathbf{x}} = \mathbf{0} \tag{1b}$$

where $\mathbf{x} \in \Re^n$ is the vector of system generalized coordinates, $\mathbf{T} \in \Re^p$ is the vector of actuator inputs, $M : \Re^n \rightarrow \Re^{n\times p}$ is bounded and of full rank, $H : \Re^n \rightarrow \Re^{n\times n}$ is the system inertia matrix, $V_{cc} : \Re^n \times \Re^n \rightarrow \Re^{n\times n}$ quantifies Coriolis and centripetal acceleration effects, $\mathbf{G} : \Re^n \rightarrow \Re^n$ arises from the system potential energy, $A : \Re^n \rightarrow \Re^{m\times n}$ is a bounded full rank matrix quantifying the nonholonomic constraints, $\lambda \in \Re^m$ is the vector of constraint multipliers, and all functions are assumed to be smooth. We note that a coordinate-free formulation could be given in terms of a mechanical control system on the tangent bundle of a smooth manifold [e.g., 26], in which case (1) would represent the system in local coordinates; however, working directly with the model (1) is sufficient for the present study. The basic problem to be considered in this investigation is that of specifying the control input $\mathbf{T}$, using only measurements of the system configuration $\mathbf{x}$ and with no knowledge of the system dynamic model (1a), so that a particular control objective is achieved. For example, a fundamental problem is the stabilization of the system configuration to some goal $\mathbf{x}_d$. It is well-known that the presence of nonholonomic constraints on the system kinematics complicates the control problem considerably. Indeed, it can be shown that for nonholonomic systems even the basic stabilization problem cannot be solved using continuous static state feedback [27,12]. These difficulties are only increased in the case of control in the presence of model and state uncertainty.

It is well-known that the mechanical system dynamics (1) possesses considerable structure. For example, for any set of generalized coordinates $\mathbf{x}$, the matrix H is symmetric and positive-definite, the matrix V_{cc} depends linearly on $\dot{\mathbf{x}}$, and the matrices H and V_{cc} are related according to $\dot{H} = V_{cc} + V_{cc}^T$ [e.g., 28]. Additionally, we will assume in what follows that the inertia matrix H and potential energy gradient $\mathbf{G}$ are bounded functions with bounded first partial derivatives; this latter property holds for virtually all mechanical systems of interest in applications, such as robotic systems and vehicles. In this case it is easy to show that V_{cc} is bounded in $\mathbf{x}$, that $V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\mathbf{y} = V_{cc}(\mathbf{x}, \mathbf{y})\dot{\mathbf{x}} \;\; \forall \mathbf{y}$, and that if $\mathbf{y}$ and $\dot{\mathbf{y}}$ are bounded then $V_{cc}(\mathbf{x}, \mathbf{y})$ is bounded and $\dot{V}_{cc}(\mathbf{x}, \mathbf{y})$ grows linearly with $\dot{\mathbf{x}}$. All of these properties will prove useful for controller development.

Observe that the rows of A, say $\mathbf{a}_i \in \Re^{1\times n}$ for $i = 1, 2, ..., m$, are smooth covectors on the configuration space $\Re^n$, and that the assumption that A is full rank implies that the codistribution spanned by the $\mathbf{a}_i$ has dimension m. The annihilator of this codistribution is an $r = n - m$ dimensional smooth distribution $\Delta = \text{span}[\mathbf{r}_1(\mathbf{x}), \mathbf{r}_2(\mathbf{x}), ..., \mathbf{r}_r(\mathbf{x})]$, where the $\mathbf{r}_i$ are smooth vector fields on the configuration space which satisfy $A\mathbf{r}_i = \mathbf{0} \;\; \forall \mathbf{x}$. Defining $R = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_r] \in \Re^{n \times r}$ permits this relationship to be expressed more concisely as $AR = 0$. As an example, let the matrix A be partitioned as $A = [A_1 \;\; A_2]$, with $A_1 \in \Re^{m \times m}$ and $A_2 \in \Re^{m \times r}$ and where A_1 is nonsingular (this is always possible, possibly with a reordering of the configuration coordinates). Then R can be constructed as follows:

$$R = \begin{bmatrix} -A_1^{-1} A_2 \\ I_r \end{bmatrix} \tag{2}$$

where I_r is the $r \times r$ identity matrix. Consider now the involutive closure of Δ, denoted Δ^* and defined as the smallest involutive distribution containing Δ. We will assume in what follows that Δ^* has constant rank n on the configuration space. In this case, Frobenius' theorem [e.g., 26] shows that the constraints are nonintegrable, so that the system (1) is (completely) nonholonomic and there is no explicit constraint on the configuration space. We note that no generality is lost with this assumption because, if the dimension of Δ^* is $m^* < n$, then $n - m*$ configuration coordinates are redundant and could be eliminated, yielding a (completely) nonholonomic system.

It is often desirable from a system control perspective to eliminate the constraint multiplier λ from the system description. One means of doing this is to first define a partition of $\mathbf{x}$ corresponding to the partition specified for A, so that $\mathbf{x} = [\mathbf{x}_1^T \;\; \mathbf{x}_2^T]^T$ with $\mathbf{x}_1 \in \Re^m$ and $\mathbf{x}_2 \in \Re^r$. Observe that the definition (2) and the constraint equation (1b) imply that the system velocities are parameterized by $\dot{\mathbf{x}}_2$ via $\dot{\mathbf{x}} = R(\mathbf{x})\dot{\mathbf{x}}_2$. This parameterization then permits (1) to be reformulated as

$$\begin{aligned} R^T M \mathbf{T} &= R^T H R \ddot{\mathbf{x}}_2 + R^T (H \dot{R} + V_{cc} R) \dot{\mathbf{x}}_2 + R^T \mathbf{G} \\ A(\mathbf{x}) \dot{\mathbf{x}} &= \mathbf{0} \end{aligned}$$

or

$$\mathbf{F} = H^*(\mathbf{x}) \ddot{\mathbf{x}}_2 + V_{cc}^*(\mathbf{x}, \dot{\mathbf{x}}_2) \dot{\mathbf{x}}_2 + \mathbf{G}^*(\mathbf{x}) \tag{3a}$$

$$\dot{\mathbf{x}}_1 = A^*(\mathbf{x}) \dot{\mathbf{x}}_2 \tag{3b}$$

where $\mathbf{F} = R^T M \mathbf{T}$, $H^* = R^T H R$, $V_{cc}^* = R^T (H\dot{R} + V_{cc} R)$, $\mathbf{G}^* = R^T \mathbf{G}$, and $A^* = -A_1^{-1} A_2$. Note that (3) is a $(2n - m)$ order differential equation which defines the evolution of the $2n - m$ system states $(\mathbf{x}, \dot{\mathbf{x}}_2)$. In what follows, it is assumed that $p \geq r$ and $R^T M$ is full rank, so that any desired $\mathbf{F}$ can be realized through proper specification of $\mathbf{T}$ and therefore the system (3a) is fully actuated. The following lemma characterizes some of the structure of (3), essentially stating that (3) inherits the "nice" structure of (1).

Lemma 1: The dynamic model terms $H^*, \mathbf{G}^*$ are bounded functions of $\mathbf{x}$ whose time derivatives $\dot{H}^*, \dot{\mathbf{G}}^*$ are also bounded in $\mathbf{x}$ and depend linearly on $\dot{\mathbf{x}}_2$, the matrix H^* is symmetric and positive-definite, and the matrices H^* and V_{cc}^* are related according to $\dot{H}^* = V_{cc}^* + V_{cc}^{*T}$. Additionally, $V_{cc}^*(\mathbf{x}, \dot{\mathbf{x}}_2)\mathbf{y} = V_{cc}^*(\mathbf{x}, \mathbf{y})\dot{\mathbf{x}}_2$ $\forall \mathbf{y}$, and if $\mathbf{y}$ and $\dot{\mathbf{y}}$ are bounded then $V_{cc}^*(\mathbf{x}, \mathbf{y})$ is bounded and $\dot{V}_{cc}^*(\mathbf{x}, \mathbf{y})$ grows linearly with $\dot{\mathbf{x}}_2$.
Proof: All of the properties can be established through direct calculation using the definitions of H^*, V_{cc}^*, and $\mathbf{G}^*$ and the properties of H, V_{cc}, and $\mathbf{G}$. ■

Each of the control systems proposed in this paper consists of two subsystems: a decentralized adaptive scheme that provides the "fictitious" control input $\mathbf{F}$ required to ensure that the given control objective is realized, and an algorithm for mapping this "fictitious" input $\mathbf{F}$ to a physically realizable control input $\mathbf{T}$. Given the assumptions regarding the matrix $R^T M$ there is always a $\mathbf{T}$ corresponding to any desired input $\mathbf{F}$, and indeed it is straightforward to determine this $\mathbf{T}$. For example, if $p = r$ then $\mathbf{T}$ is uniquely determined from $\mathbf{F}$ via $\mathbf{T} = (R^T M)^{-1}\mathbf{F}$, and if $p > r$ then $\mathbf{T}$ can be chosen to give the desired $\mathbf{F}$ and simultaneously achieve some additional objective. Therefore for the remainder of the paper it is assumed that $\mathbf{F}$ can be commanded directly, and the focus of the subsequent discussion is on specifying this input so that the given control objective is realized in the presence of model and state uncertainty. In this paper we shall address both the motion stabilization problem, in which the objective is to ensure that the system evolves from its initial state to some desired final state, and the trajectory tracking problem, in which the evolution from initial state to desired final state is to be along some specified trajectory. In what follows, it is assumed that $\mathbf{x}$ (but not $\dot{\mathbf{x}}$) is measurable and that little information is available regarding the system model.

3. Adaptive Stabilization to an Equilibrium Manifold

In this section, we begin our study of the problem of stabilizing the nonholonomic system (1) in the presence of uncertainty. Recall that the nonholonomic nature of the constraints significantly increases the difficulty of controlling this system, so that even the basic problem of stabilizing (1) to a desired (constant) configuration $\mathbf{x}_d$ is not solvable using continuous static state feedback [27,12]. In view of this fact, we examine first the simpler problem of stabilizing (1) to an "equilibrium manifold" of dimension m rather than to an equilibrium point (see, e.g., [12] for a definition of stabilizing a system to a manifold). Note that this problem is solvable using smooth static feedback and can be viewed as the natural extension of work on stabilizing holonomically constrained systems.

3.1 Control Scheme

Let us define an equilibrium manifold N_e as follows [12]:

$$N_e = \{\mathbf{x}, \dot{\mathbf{x}} \mid \mathbf{s}(\mathbf{x}) = \mathbf{0}, \dot{\mathbf{x}} = \mathbf{0}\}$$

where $\mathbf{s} : \Re^n \rightarrow \Re^r$ is a smooth "output" function which can be specified as desired subject to the (transversality) condition that the matrices $\partial\mathbf{s}/\partial\mathbf{x}_2$ and $(\partial\mathbf{s}/\partial\mathbf{x})R$ are nonsingular. We seek a control law which stabilizes (1) to the smooth m-dimensional manifold N_e and which is implementable without any knowledge of the system dynamic model and using only measurements of system configuration. For simplicity of development we shall choose $\mathbf{s} = \mathbf{x}_2$ in what follows, so that the system representation (3) can be used directly. However, a system model analogous to (3) can be derived in terms of any other output $\mathbf{s}$ which satisfies the transversality condition, so that the controller design procedure given here can be applied in the general case [29]. In fact, the model independence of the proposed approach to system stabilization makes the controller development quite modular and easy to apply for any choice of output.

Consider the following adaptive scheme:

$$\begin{aligned} F_i &= f_i(t) + k_1\gamma^2 w_i + k_2\gamma^2 e_{si} \\ \dot{w}_i &= -2\gamma w_i + \gamma^2 \dot{e}_{si} \\ \dot{f}_i &= -\sigma f_i + \beta q_{si} \qquad \text{for} \quad i = 1, 2, ..., n-m \end{aligned} \tag{4}$$

where the subscript i refers to the ith element of a vector, $\mathbf{e}_s = -\mathbf{s}$ is a measure of distance from N_e, $\mathbf{q}_s = \dot{\mathbf{e}}_s + k_2\mathbf{e}_s/k_1\gamma - \mathbf{w}/\gamma$, $\mathbf{f}(t) \in \Re^r$ is the adaptive element, k_1, k_2, γ are positive constants, β is the adaptation gain for $\mathbf{f}$, and σ is defined as follows: $\sigma = 0$ if $\| \mathbf{f} \| \leq f_{max}$ and $\sigma = \sigma_0$ otherwise, with σ_0 and f_{max} positive scalars. The scheme (4) can be viewed as a natural extension of the well-known PD-plus-potential-energy control law for mechanical systems [e.g., 26], with the velocity feedback component of the controller replaced by a first order linear compensator and with potential energy compensation provided through adaptation. It will be seen below that $\mathbf{w}$ provides a simple means of injecting damping into the closed-loop system without using velocity feedback, and that $\mathbf{f}$ effectively compensates for the potential energy effects represented by $\mathbf{G}^*$. The definition of σ provides a "switching σ-modification" adaptation law [30]. This approach to adaptive control has been proposed for linear systems to provide robust performance in the presence of external disturbances while retaining asymptotic convergence of the system error in the ideal case in which no disturbances are present [30]. In what follows, it will be shown that the formulation (4) yields a similar robust adaptive performance for the problem of stabilizing the nonlinear mechanical system (1). Observe that (4) is implementable without velocity information because, although $\dot{\mathbf{w}}$ and $\dot{\mathbf{f}}$ depend on $\dot{\mathbf{e}}_s$, the control law terms $\mathbf{w}$ and $\mathbf{f}$ depend only on $\mathbf{e}_s$ (as can be verified through direct integration of the $\dot{\mathbf{w}}$ and $\dot{\mathbf{f}}$ equations).

The stability properties of the proposed adaptive regulation strategy (4) are summarized in the following theorem.

Theorem 1: If γ is chosen sufficiently large then the adaptive scheme (4) ensures that (3) evolves with $\mathbf{e}_s$, $\dot{\mathbf{e}}_s$, $\mathbf{w}$, and $\mathbf{f}$ (semiglobally) uniformly bounded and so that:

1. If the system potential energy $U(\mathbf{x})$ and its first partials are constant on N_e then $\mathbf{e}_s$ and $\dot{\mathbf{e}}_s$ converge asymptotically to zero (note that potential energy functions of this sort are very common in applications).
2. If the system potential energy $U(\mathbf{x})$ varies on N_e then $\mathbf{e}_s$ and $\dot{\mathbf{e}}_s$ converge asymptotically to a compact set which can be made arbitrarily small (by increasing β).

Proof: We consider first the case in which $U(\mathbf{x})$ and its first partials are constant on N_e. Applying the control law (4) to the mechanical system dynamics (3) yields the closed-loop error dynamics

$$H^*\ddot{\mathbf{e}}_s + V_{cc}^*\dot{\mathbf{e}}_s + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e}_s + \mathbf{\Phi} + \mathbf{G}^*(\mathbf{0}) - \mathbf{G}^*(\mathbf{x}) = \mathbf{0} \tag{5}$$

where $\mathbf{\Phi} = \mathbf{f} - \mathbf{G}^*(\mathbf{0})$. Consider the Lyapunov function candidate

$$\begin{aligned} V_1 = {} & \frac{1}{2}\dot{\mathbf{e}}_s^T H^*\dot{\mathbf{e}}_s + \frac{1}{2}k_2\gamma^2\mathbf{e}_s^T\mathbf{e}_s + \frac{1}{2}k_1\mathbf{w}^T\mathbf{w} + \frac{k_2}{k_1\gamma}\mathbf{e}_s^T H^*\dot{\mathbf{e}}_s \\ & - \frac{1}{\gamma}\mathbf{w}^T H^*\dot{\mathbf{e}}_s + \frac{1}{2\beta}\mathbf{\Phi}^T\mathbf{\Phi} + U(\mathbf{x}) - U(\mathbf{0}) + \mathbf{G}^{*T}(\mathbf{0})\mathbf{e}_s \end{aligned} \tag{6}$$

and note that V_1 is a positive-definite and radially-unbounded function of $\mathbf{e}_s$, $\dot{\mathbf{e}}_s$, $\mathbf{w}$ and $\mathbf{\Phi}$ if γ is chosen sufficiently large (for example, if γ is chosen large enough then the term $k_2\gamma^2\mathbf{e}_s^T\mathbf{e}_s/2 + U(\mathbf{x}) - U(\mathbf{0}) + \mathbf{G}^{*T}(\mathbf{0})\mathbf{e}_s$ can be shown to be positive-definite in $\mathbf{e}_s$).

Differentiating (6) along (5) and simplifying using the structural properties of the mechanical system dynamics summarized in Lemma 1 yields

$$\begin{aligned} \dot{V}_1 = {} & -(\gamma - \frac{k_2}{k_1\gamma})\dot{\mathbf{e}}^T H^*\dot{\mathbf{e}} - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 + 2\mathbf{w}^T H^*\dot{\mathbf{e}} \\ & + \frac{1}{\gamma}\dot{\mathbf{e}}^T[\frac{k_2}{k_1}V_{cc}^*\mathbf{e} - V_{cc}^*\mathbf{w}] + \frac{1}{\beta}\mathbf{\Phi}^T(\dot{\mathbf{f}} - \beta\mathbf{q}_s) + (\mathbf{G}^*(\mathbf{x}) - \mathbf{G}^*(\mathbf{0}))(\frac{k_2}{k_1\gamma}\mathbf{e}_s - \frac{1}{\gamma}\mathbf{w}) \\ \leq {} & -(\gamma - \frac{k_2}{k_1\gamma})\lambda_m(H^*) \parallel \dot{\mathbf{e}}_s \parallel^2 - (\frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma}) \parallel \mathbf{e}_s \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\ & + 2\lambda_M(H^*) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}}_s \parallel + \frac{M}{\gamma} \parallel \mathbf{w} \parallel\parallel \mathbf{e}_s \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e}_s \parallel\parallel \dot{\mathbf{e}}_s \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}}_s \parallel^2 \end{aligned} \tag{7}$$

where $\lambda_m(\cdot), \lambda_M(\cdot)$ denote the minimum and maximum eigenvalue of the matrix argument, respectively, k_{cc} is a scalar upper bound on the linear dependency of V_{cc}^* on $\dot{\mathbf{x}}$ (i.e., $\| V_{cc}^* \|_F \leq k_{cc} \| \dot{\mathbf{x}} \| \; \forall \mathbf{x}$ with $\| \cdot \|_F$ the Frobenius matrix norm), and M is a positive scalar constant satisfying $M \| \mathbf{x} \| \geq \| \mathbf{G}^*(\mathbf{x}) - \mathbf{G}^*(\mathbf{0}) \| \; \forall \mathbf{x}$ (the boundedness of the partial derivatives of $\mathbf{G}^*$ ensures that such an M exists). Define $\mathbf{z} = [\| \mathbf{e}_s \| \;\; \| \dot{\mathbf{e}}_s \| \;\; \| \mathbf{w} \|]^T$ and Q_1 as

$$Q_1 = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma} & 0 & -\frac{M}{2\gamma} \\ 0 & (\gamma - \frac{k_2}{k_1\gamma})\lambda_m(H^*) & -\lambda_M(H^*) \\ -\frac{M}{2\gamma} & -\lambda_M(H^*) & k_1\gamma \end{bmatrix}$$

and note that Q_1 is positive-definite if γ is chosen large enough. Then the following bound on $\dot{V}_1$ in (7) can be obtained through routine manipulation:

$$\dot{V}_1 \leq -(\lambda_m(Q_1) - \frac{\alpha_1}{\gamma} {V_1}^{1/2}) \| \mathbf{z} \|^2 \tag{8}$$

where α_1 is a scalar constant which does not increase as γ is increased. Let $c_1(t) = \lambda_m(Q_1) - \alpha_1 {V_1}^{1/2}(t)/\gamma$ and choose γ large enough so that $c_1(0) > 0$ (this is always possible). Then (8) implies that $c_1(0) \leq c_1(t) \;\; \forall t$, so that $\dot{V}_1 \leq -c_1(0) \| \mathbf{z} \|^2$. Standard arguments can then be used to show that $\mathbf{z}$ and $\mathbf{\Phi}$ are bounded (and therefore that $\mathbf{e}_s$, $\dot{\mathbf{e}}_s$, $\mathbf{w}$, and $\mathbf{f}$ are bounded) and that $\mathbf{z}$ converges to zero (and therefore that $\mathbf{e}_s$, $\dot{\mathbf{e}}_s$, and $\mathbf{w}$ converge to zero).

We consider next the case of general $U(\mathbf{x})$. Applying the adaptive scheme (4) to the mechanical system dynamics (3) yields the closed-loop error dynamics

$$H^* \ddot{\mathbf{e}}_s + V_{cc}^* \dot{\mathbf{e}}_s + k_1\gamma^2 \mathbf{w} + k_2\gamma^2 \mathbf{e}_s + \mathbf{\Phi}_d = \mathbf{0} \tag{9}$$

where $\mathbf{\Phi}_d = \mathbf{f} - \mathbf{G}^*$. Define the Lyapunov function candidate

$$V_2 = \frac{1}{2}\dot{\mathbf{e}}_s^T H^* \dot{\mathbf{e}}_s + \frac{1}{2}k_2\gamma^2 \mathbf{e}_s^T \mathbf{e}_s + \frac{1}{2}k_1 \mathbf{w}^T \mathbf{w} + \frac{k_2}{k_1\gamma}\mathbf{e}_s^T H^* \dot{\mathbf{e}}_s - \frac{1}{\gamma}\mathbf{w}^T H^* \dot{\mathbf{e}}_s + \frac{1}{2\beta}\mathbf{\Phi}_d^T \mathbf{\Phi}_d \tag{10}$$

and note that V_2 is a positive-definite and radially-unbounded function of $\mathbf{e}_s$, $\dot{\mathbf{e}}_s$, $\mathbf{w}$ and $\mathbf{\Phi}_d$ if γ is chosen sufficiently large. Differentiating (10) along (9) and simplifying as in the first part of the proof yields

$$\begin{aligned} \dot{V}_2 \leq & -(\gamma - \frac{k_2}{k_1\gamma})\lambda_m(H^*) \| \dot{\mathbf{e}}_s \|^2 - \frac{k_2^2\gamma}{k_1} \| \mathbf{e}_s \|^2 - k_1\gamma \| \mathbf{w} \|^2 + 2\lambda_M(H^*) \| \mathbf{w} \| \| \dot{\mathbf{e}}_s \| \\ & + \frac{k_2 k_{cc}}{k_1\gamma} \| \mathbf{e}_s \| \| \dot{\mathbf{e}}_s \|^2 + \frac{k_{cc}}{\gamma} \| \mathbf{w} \| \| \dot{\mathbf{e}}_s \|^2 + \frac{1}{\beta}\mathbf{\Phi}_d^T(\dot{\mathbf{f}} - \dot{\mathbf{G}}^* - \beta \mathbf{q}_s) \end{aligned} \tag{11}$$

Defining Q_2 as

$$Q_2 = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} & 0 & 0 \\ 0 & (\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_m(H^*) & -\lambda_M(H^*) \\ 0 & -\lambda_M(H^*) & k_1\gamma \end{bmatrix}$$

(and noticing that Q_2 is positive-definite if γ is chosen large enough) permits the following bound on $\dot{V}_2$ in (11) to be obtained:

$$\dot{V}_2 \leq -\mathbf{z}^T Q_2 \mathbf{z} + \frac{k_2 k_{cc}}{k_1 \gamma} \parallel \mathbf{e}_s \parallel \parallel \mathbf{e}_s \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel \parallel \dot{\mathbf{e}}_s \parallel^2$$
$$- \frac{\gamma}{2}\lambda_m(H^*) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{1}{\beta}\mathbf{\Phi}_d^T(\sigma \mathbf{f} + \dot{\mathbf{G}}^*) \tag{12}$$

If f_{max} is selected so that $\parallel \mathbf{G}^*(\mathbf{x}) \parallel \leq f_{max} \ \ \forall \mathbf{x}$ then the following bound on $\dot{V}_2$ in (12) can be derived through routine manipulation:

$$\dot{V}_2 \leq -(\lambda_m(Q_2) - \frac{\alpha_2}{\gamma}{V_2}^{1/2}) \parallel \mathbf{z} \parallel^2 - \frac{\sigma}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2 + \frac{\eta_2}{\beta}$$

where α_2 and η_2 are scalar constants which do not increase as γ, β increase.

The remainder of the proof will utilize the ultimate boundedness result presented by the authors in [23,24], which is based on the work of Corless and Leitmann [31]. Toward this end, it is convenient to bound V_2 in (10) in the following manner:

$$\lambda_1 \parallel \mathbf{z} \parallel^2 + \frac{1}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2 \leq V_2 \leq \lambda_3 \parallel \mathbf{z} \parallel^2 + \frac{1}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2$$

where the λ_i are positive scalar constants independent of β. Now choose two scalar constants V_M, V_m so that $V_M > V_m \geq V_2(0)$, and define $c_M = \lambda_m(Q_2) - \alpha_2 {V_M}^{1/2}/\gamma$; then choose γ large enough so that $c_M > 0$ (this is always possible). Let $\delta^* = \lambda_3\eta_2/c_M + 2\max(2f_{max}^2, \eta_2/\sigma_0)$ and choose β_0 so that $\delta^*/\beta_0 < V_m$ (this is always possible). Then selecting $\beta \geq \beta_0$ ensures that if $V_m \leq V_2 \leq V_M$ then $\dot{V}_2 < 0$. This condition together with $V_M > V_m \geq V_2(0)$ implies that $V_2(t) \leq V_M \ \forall t$, so that $c_2(t) = \lambda_m(Q_2) - \alpha_2 {V_2}^{1/2}(t)/\gamma > c_M > 0 \ \ \forall t$ and

$$\dot{V}_2 \leq -c_M \parallel \mathbf{z} \parallel^2 - \frac{\sigma}{2(\beta_0 + \Delta\beta)} \parallel \mathbf{\Phi}_d \parallel^2 + \frac{\eta_2}{(\beta_0 + \Delta\beta)}$$

where $\Delta\beta = \beta - \beta_0$ and it is assumed that β is chosen so that $\Delta\beta > 0$. It follows immediately that $\parallel \mathbf{z} \parallel$ and $\parallel \mathbf{\Phi}_d \parallel$ are uniformly bounded. Moreover, the ultimate boundedness result given in [23,24] is now directly applicable and permits the conclusion that $\parallel \mathbf{z} \parallel$ converges asymptotically to a closed ball of radius $r^2 = \delta^*/(\lambda_1(\beta_0 + \Delta\beta))$. ■

Several observations can be made concerning the decentralized adaptive stabilization strategy (4). First, the preceding stability analysis demonstrates that this scheme ensures that the mechanical system dynamics (3) evolves from any initial configuration to any desired (and suitably defined) m-dimensional equilibrium manifold with uniformly bounded signals and with desirable convergence properties for the output error: asymptotic convergence to zero if the system potential energy is a constant on the equilibrium manifold and convergence to an arbitrarily small neighborhood of zero in the general case. Note that, in the general case, the ultimate size of the regulation error can be reduced as desired simply by increasing a single controller parameter (the adaptation gain β). It is easy to show that the above robustness to potential energy variation on N_e implies a robustness to *any* bounded external disturbance; this generalization to the case of external disturbances is analogous to that given in [23,24] for holonomically constrained robotic systems and is worked out in detail in [29]. The adaptive stabilization strategy (4) possesses a simple decentralized structure, utilizes only measurements of system configuration, and does not require knowledge of the structure or parameter values for the system model or any payload. Thus the proposed scheme provides a computationally efficient, modular, and readily implementable solution to the equilibrium manifold stabilization problem. Observe that the stability properties established in Theorem 1 are *semiglobal* in the sense that the region of attraction can be increased arbitrarily by increasing the controller gain γ. It is stressed that this does not imply that γ must be chosen to be overly large; indeed, we have found that excellent performance is obtainable with this gain set to quite modest values. It is mentioned that, for clarity of exposition, the calculations required in the proof of the Theorem 1 have been condensed significantly in the above presentation; the interested reader is referred to the report [29] for the details of all of these calculations. An additional observation is that the scalar controller gains k_1, k_2, γ, and β can be replaced with the appropriate diagonal matrices for increased flexibility and performance; scalar gains are used above for simplicity of development.

3.2 Implementation Results

The adaptive stabilization scheme described above is now applied to a mobile robot through computer simulation. The mobile robot chosen for this simulation study is the two wheel vehicle shown in Figure 1 and described in [14]. The dynamic model (1) has the following form for this system:

$$\begin{aligned} m\ddot{x} &= \lambda\cos\theta - p_1(T_1 + T_2)\sin\theta + d_1(t) \\ m\ddot{y} &= \lambda\sin\theta + p_1(T_1 + T_2)\cos\theta + d_2(t) \qquad (13) \\ J\ddot{\theta} &= p_2(T_1 - T_2)\ {}^{20} \end{aligned}$$

$$0 = \dot{x}\cos\theta + \dot{y}\sin\theta$$

where x, y, θ are the position and orientation coordinates for the (axle of the) mobile robot, m, J are the system inertial parameters, p_1, p_2 are kinematic parameters, T_1, T_2 are the torques provided at the wheels, and d_1, d_2 are external disturbances. In this simulation, d_1, d_2 are modeled as the sum of a constant (bias) force with magnitude equal to one half the maximum undisturbed control force and a zero-mean Gaussian signal with standard deviation of one tenth the maximum undisturbed control force. Note that in all of the simulations presented in this paper the unit of length is meter, the unit of time is second, and the unit of angle is degree, unless stated otherwise.

In this simulation study, the vector of system outputs is chosen to be $\mathbf{s} = [y \quad \theta]^T$; thus the control objective is to drive the mobile robot from any initial configuration $\mathbf{x}(0) = [x(0) \quad y(0) \quad \theta(0)]^T$ to the x-axis in the presence of external disturbances and with no model information or rate measurements. The stabilization scheme (4) is applied to the mathematical model of the mobile robot described above and shown in Figure 1 through computer simulation with a sampling period of two milliseconds. All integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of two milliseconds. Additional details concerning the simulation strategy for this study can be found in [29]. The system model parameters are defined as $m = J = 10$ and $p_1 = p_2 = 1$, and the controller parameters k_1, k_2, γ are set as follows: $k_1 = 10$, $k_2 = 20$, $\gamma = 5$. The adaptive gain $\mathbf{f}$ is set to zero initially and the adaptation parameters are set as follows: $\sigma_0 = 0.1$, $\beta = 10$, and $f_{max} = 100$. It is noted that no attempt was made to "tune" the gains to obtain the best possible performance. The stabilization scheme (4) was tested using a wide range of initial conditions. Results for a typical simulation are given in Figures 2a and 2b, with Figure 2a showing the response of the y coordinate and Figure 2b depicting the evolution of the orientation coordinate. The response for other initial conditions were quite similar and hence are not shown. Observe that accurate and robust equilibrium manifold stabilization of the system is achieved. We mention that other choices of system output and external disturbances were also tested in this simulation study and that similar performance was obtained in all cases.

4. Adaptive Control of Differentially Flat Systems

Having developed a simple and robust strategy for stabilizing the nonholonomic mechanical system (1) (or (3)) to an m-dimensional equilibrium manifold, we now consider the problem of stabilizing the system to an equilibrium point $\mathbf{x}_d$. As mentioned above, it is known that this problem is not solvable using continuous static state feedback [27,12]. However, it has been shown in [12] that the system (1) is strongly accessible and small time locally controllable at an equilibrium state $(\mathbf{x}_d, \mathbf{0})$. Thus it makes sense to study the equilibrium point stabilization problem, but to investigate techniques other than the simple strategy derived in the previous section. In this section and the next we focus our attention on important subclasses of the nonholonomic mechanical system (1), and develop strategies for controlling these systems which exploit the additional structure present in these subclasses. In each case our approach is to first consider the problem of generating a trajectory for the system which will take the system to the goal configuration, and then to design a controller capable of ensuring that the system accurately tracks this trajectory in the presence of model and measurement uncertainty.

4.1 Background

One type of nonholonomic system for which trajectory generation is particularly straightforward is the class of so-called *differentially flat systems* [e.g., 7,17]. Roughly speaking, a system is differentially flat if there is a set of outputs, equal in number to the number of inputs, such that all states and inputs can be determined from these outputs without integration. More precisely, a system with states $\mathbf{s} \in \Re^{2n}$ and inputs $\mathbf{u} \in \Re^{p}$ is flat if we can find outputs $\mathbf{y} \in \Re^{p}$ of the form

$$\mathbf{y} = \mathbf{y}(\mathbf{s}, \mathbf{u}, \dot{\mathbf{u}}, ..., \mathbf{u}^{(q)})$$

such that

$$\mathbf{s} = \mathbf{s}(\mathbf{y}, \dot{\mathbf{y}}, ..., \mathbf{y}^{(s)}) \tag{14a}$$

$$\mathbf{u} = \mathbf{u}(\mathbf{y}, \dot{\mathbf{y}}, ..., \mathbf{y}^{(s)}) \tag{14b}$$

Differentially flat systems have attracted considerable attention recently and, while there are no general methods for determining whether or not a particular system is flat, it is known that many systems of interest in applications are flat. For example, all of the following are flat: mobile robots and cars, a car pulling N (properly hitched) trailers, hopping robots, underwater vehicles, chains of planar rigid bodies, and planar satellite/manipulator systems. Differentially flat systems were originally studied in the context of differential algebra [e.g., 7] and more recently using more traditional geometric control methods [e.g., 17]. While differential flatness can be viewed as a generalization of dynamic feedback linearization, it is in some ways more natural to view flatness in terms of trajectory generation. This perspective is the one taken here, and our approach shall be to exploit flatness to solve the problem of finding a feasible trajectory from the initial configuration to the final desired configuration $\mathbf{x}_d$, and then to design a control strategy to track this desired trajectory in the presence of uncertainty. 2 - 22

4.2 Control Scheme

Consider the problem of causing the mechanical system (1) with m nonholonomic constraints and $r = n - m$ inputs to evolve from its initial state $(\mathbf{x}(0), \dot{\mathbf{x}}_2(0))$ to a specified goal state $(\mathbf{x}_d, \mathbf{0})$ during some time interval $[0, T]$. Suppose that a vector of flat outputs $\mathbf{y} \in \Re^r$ has been identified. Note that, from (14a), there is a set of outputs and derivatives of outputs $(\mathbf{y}(0), \dot{\mathbf{y}}(0), ..., \mathbf{y}^{(s)}(0))$ corresponding to the initial state and another set $(\mathbf{y}(T), \dot{\mathbf{y}}(T), ..., \mathbf{y}^{(s)}(T))$ corresponding to the goal state. The trajectory generation problem can therefore be reduced to the problem of finding a curve in the flat output space with the required initial and final location, slope, curvature, and so on. This curve fitting problem is standard and can be solved in a number of ways, and any solution to this simple problem then defines a feasible trajectory for the full system (1) which produces the requisite motion.

Observe that we now have a means of defining a trajectory for the r-dimensional flat output $\mathbf{y}$ which ensures that the nonholonomic system evolves as desired, and we also have a strategy for controlling the r-dimensional configuration vector $\mathbf{x}_2$ associated with the "reduced" system (3a). Suppose that a matrix R could be found for which $\dot{\mathbf{x}} = R(\mathbf{x})\dot{\mathbf{y}}$. In this case the system (1) could be reformulated in such a way that the flat output $\mathbf{y}$ and the "reducing" output $\mathbf{x}_2$ are equal, and then a natural and direct means of controlling the system would be to first specify the desired trajectory $\mathbf{y}_d(t)$ for the flat output and then generate the control input $\mathbf{F}$ which ensures that $\mathbf{x}_2 (= \mathbf{y})$ in (3a) tracks this trajectory. Unfortunately, the next lemma shows that the flat and reducing outputs do not coincide except in the trivial case of a holonomically constrained system.

Lemma 2: If the system (1) admits a flat output $\mathbf{y}$ and a reducing output $\mathbf{x}_2$, then $\mathbf{y} = \mathbf{x}_2$ only if the system constraints are integrable.

Proof: Suppose that $\mathbf{x}_2$ is a vector of outputs which is both reducing and flat. Then the configuration coordinates $\mathbf{x} = [\mathbf{x}_1^T \;\; \mathbf{x}_2^T]^T$ must satisfy $\dot{\mathbf{x}}_1 = R_1(\mathbf{x}_1, \mathbf{x}_2)\dot{\mathbf{x}}_2$ for some $R_1 \in \Re^{m \times r}$ and also $\mathbf{x}_1 = \mathbf{g}(\mathbf{x}_2, ..., \mathbf{x}_2^{(s)})$ for some function $\mathbf{g}$ and integer s. Direct differentiation of $\mathbf{x}_1 = \mathbf{g}(\mathbf{x}_2, ..., \mathbf{x}_2^{(s)})$ shows that $\dot{\mathbf{x}}_1$ depends on derivatives of $\mathbf{x}_2$ through order $s+1$, while the relationship $\dot{\mathbf{x}}_1 = R_1(\mathbf{g}_1(\mathbf{x}_2, ..., \mathbf{x}_2^{(s)}), \mathbf{x}_2)\dot{\mathbf{x}}_2$ indicates that $\dot{\mathbf{x}}_1$ depends on derivatives of $\mathbf{x}_2$ through order $\max(1, s)$. Taken together these results imply that the only way $\mathbf{x}_2$ can be both reducing and flat is if $s = 0$, but in that case $\mathbf{x}_1 = \mathbf{g}(\mathbf{x}_2)$ and the constraints are integrable. ■

Lemma 2 shows that for nonholonomic systems of the form (1), the flat and reducing outputs cannot coincide. As a consquence, it is not possible to simply generate a trajectory for the flat outputs which produces the desired motion and then track this trajectory using the reduced model (3a). However, it is possible to adopt essentially this strategy by adding one additional step in the algorithm. Thus we propose the following solution to the equilibrium point stabilization problem for differentially flat systems:

Algorithm 1:

1. Given an initial state $(\mathbf{x}(0), \dot{\mathbf{x}}_2(0))$ and a goal state $(\mathbf{x}_d, \mathbf{0})$, determine the sets of flat outputs and derivatives of outputs $(\mathbf{y}(0), \dot{\mathbf{y}}(0), ..., \mathbf{y}^{(s)}(0))$ and $(\mathbf{y}(T), \dot{\mathbf{y}}(T), ..., \mathbf{y}^{(s)}(T))$ corresponding to the initial and goal states. Find a smooth curve $\mathbf{y}_d(t)$ in the flat output space connecting these initial and final output values.

2. Use the relation $\mathbf{x}_{2d}(t) = \mathbf{g}(\mathbf{y}_d(t), ..., \mathbf{y}_d^{(s)}(t))$ to determine the desired trajectory for $\mathbf{x}_2$. Observe that this trajectory completely specifies the velocity of the nonholonomic system (1) at each instant and therefore ensures that the full system state evolves as desired.

3. Track the desired trajectory $\mathbf{x}_{2d}(t)$ for the configuration $\mathbf{x}_2$ of the reduced system (3a) through proper specification of the control input $\mathbf{F}$.

In order to successfully implement the proposed approach to equilibrium point stabilization, it is necessary to ensure that the (reduced) mechanical system (3a) closely tracks the desired trajectory $\mathbf{x}_{2d}$ (where it is assumed that $\mathbf{x}_{2d}$ is bounded with bounded derivatives). Our objective is to develop a control scheme which is capable of accurate tracking control without rate measurements or knowledge of the system dynamic model. Consider the decentralized adaptive tracking controller obtained from (4) through the introduction of feedforward elements in the control law:

$$\begin{aligned} F_i &= a_i(t)\ddot{x}_{2di} + b_i(t)\dot{x}_{2di} + f_i(t) + k_1\gamma^2 w_i + k_2\gamma^2 e_i \\ \dot{w}_i &= -2\gamma w_i + \gamma^2 \dot{e}_i \qquad \text{for} \quad i = 1, 2, ..., r \end{aligned} \tag{15}$$

where $\mathbf{e} = \mathbf{x}_{2d} - \mathbf{x}_2$ and the $a_i(t)$ and $b_i(t)$ are (feedforward) adaptive gains which are adjusted according to the following simple update laws (the adaptation laws for the $f_i(t)$ are repeated here for convenience of reference):

$$\begin{aligned} \dot{f}_i &= -\sigma_1 f_i + \beta_1 q_i \\ \dot{a}_i &= -\sigma_2 a_i + \beta_2 q_i \ddot{x}_{2di} \\ \dot{b}_i &= -\sigma_3 b_i + \beta_3 q_i \dot{x}_{2di} \qquad \text{for} \quad i = 1, 2, ..., r \end{aligned} \tag{16}$$

where $\mathbf{q} = \dot{\mathbf{e}} + k_2\mathbf{e}/k_1\gamma - \mathbf{w}/\gamma$ represents a weighted and filtered error term and the σ_j and β_j are positive scalar adaptation gains. Observe that the scheme (15),(16) is implementable without velocity information because, although $\dot{\mathbf{w}}$, $\dot{\mathbf{f}}$, $\dot{\mathbf{a}}$, and $\dot{\mathbf{b}}$ depend on $\dot{\mathbf{e}}$, the control law requires the terms $\mathbf{w}$, $\mathbf{f}$, $\mathbf{a}$, and $\mathbf{b}$ and these can be integrated so as to depend only on $\mathbf{e}$.

The stability properties of the proposed decentralized adaptive tracking controller (15),(16) are summarized in the following theorem.

Theorem 2: The adaptive controller (15),(16) ensures that $\mathbf{e}, \dot{\mathbf{e}}, \mathbf{w}, \mathbf{f}, \mathbf{a}, \mathbf{b}$ are semiglobally uniformly bounded provided γ is chosen sufficiently large. Moreover, the trajectory tracking error $\mathbf{e}, \dot{\mathbf{e}}$ is guaranteed to converge exponentially to a compact set which can be made arbitrarily small.

Proof: Applying the control law (15) to the mechanical system (3a) yields the closed-loop system dynamics

$$H^*\ddot{\mathbf{e}} + V_{cc}^*\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi}_f + [\mathbf{a}\ddot{\mathbf{x}}_{2d}] + [\mathbf{b}\dot{\mathbf{x}}_{2d}] + V_{cd}\dot{\mathbf{e}} = \mathbf{0} \tag{17}$$

where the notation $[\mathbf{st}] = [s_1t_1, s_2t_2, ..., s_nt_n]^T \in \Re^n$ (for any two n-vectors $\mathbf{s}$ and $\mathbf{t}$) is introduced, $V_{cd} = V_{cc}^*(\mathbf{x}, \dot{\mathbf{x}}_{2d})$, and $\mathbf{\Phi}_f = \mathbf{f} - \mathbf{G}^* - H^*\ddot{\mathbf{x}}_{2d} - V_{cd}\dot{\mathbf{x}}_{2d}$. Note that in obtaining (17) the Coriolis/centripetal term was expanded as follows: $V_{cc}^*\dot{\mathbf{x}}_{2d} = V_{cd}\dot{\mathbf{x}}_2 = V_{cd}\dot{\mathbf{x}}_{2d} - V_{cd}\dot{\mathbf{e}}$. Consider the Lyapunov function candidate

$$V_3 = V_2 + \frac{1}{2\beta_2}\mathbf{a}^T\mathbf{a} + \frac{1}{2\beta_3}\mathbf{b}^T\mathbf{b} \tag{18}$$

where it is understood that V_2 is now a function of $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{\Phi}_f$ rather then $\mathbf{e}_s$, $\dot{\mathbf{e}}_s$, $\mathbf{\Phi}_d$. Note that V_3 is a positive-definite and radially-unbounded function of $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$, $\mathbf{\Phi}_f$, $\mathbf{a}$, and $\mathbf{b}$ if γ is chosen sufficiently large. Computing the derivative of (18) along (17) and simplifying using the adaptation laws (16) yields

$$\begin{aligned}
\dot{V}_3 = &-(\gamma - \frac{k_2}{k_1\gamma})\dot{\mathbf{e}}^T H^*\dot{\mathbf{e}} - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 + 2\mathbf{w}^T H^*\dot{\mathbf{e}} \\
&+ \frac{1}{\gamma}\dot{\mathbf{e}}^T[\frac{k_2}{k_1}V_{cc}^*\mathbf{e} - V_{cc}^*\mathbf{w}] - \mathbf{q}^T V_{cd}\dot{\mathbf{e}} + \frac{1}{\beta_1}\mathbf{\Phi}_f^T(\dot{\mathbf{\Phi}}_f - \beta_1\mathbf{q}) \\
&+ \frac{1}{\beta_2}\mathbf{a}^T(\dot{\mathbf{a}} - \beta_2[\mathbf{q}\ddot{\mathbf{x}}_{2d}]) + \frac{1}{\beta_3}\mathbf{b}^T(\dot{\mathbf{b}} - \beta_3[\mathbf{q}\dot{\mathbf{x}}_{2d}]) \\
\leq &-((\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_m(H^*) - k_{cd}) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\
&+ (2\lambda_M(H^*) + \frac{k_{cd}}{\gamma} + \frac{k_{cc}v_{max}}{\gamma}) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{k_2}{k_1\gamma}(k_{cc}v_{max} + k_{cd}) \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel \\
&+ \frac{k_2k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma_2}{\beta_2} \parallel \mathbf{a} \parallel^2 - \frac{\sigma_3}{\beta_3} \parallel \mathbf{b} \parallel^2 \\
&- \frac{1}{\beta_1}\mathbf{\Phi}_f^T(\sigma_1\mathbf{f} + \frac{d}{dt}(\mathbf{G}^* + H^*\ddot{\mathbf{x}}_{2d} + V_{cd}\dot{\mathbf{x}}_{2d}))
\end{aligned} \tag{19}$$

where k_{cd} is an upper bound on V_{cd} (i.e., $\| V_{cd} \|_F \leq k_{cd} \;\; \forall \mathbf{x}$) and v_{max} is an upper bound on $\| \dot{\mathbf{x}}_{2d} \|$. Manipulations similar to those used in the proof of Theorem 1 then permit the following upper bound on $\dot{V}_3$ in (19) to be established:

$$\begin{aligned} \dot{V}_3 \leq & -((\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_m(H^*) - k_{cd}) \| \dot{\mathbf{e}} \|^2 - \frac{k_2^2\gamma}{k_1} \| \mathbf{e} \|^2 - k_1\gamma \| \mathbf{w} \|^2 \\ & + (2\lambda_M(H^*) + \frac{k_{cd}}{\gamma} + \frac{k_{cc}v_{max}}{\gamma}) \| \mathbf{w} \| \| \dot{\mathbf{e}} \| \\ & + \frac{k_2}{k_1\gamma}(k_{cc}v_{max} + k_{cd}) \| \mathbf{e} \| \| \dot{\mathbf{e}} \| + \frac{k_2 k_{cc}}{k_1\gamma} \| \mathbf{e} \| \| \dot{\mathbf{e}} \|^2 + \frac{k_{cc}}{\gamma} \| \mathbf{w} \| \| \dot{\mathbf{e}} \|^2 \\ & - \frac{\sigma_1}{2\beta_1} \| \mathbf{\Phi}_f \|^2 - \frac{\sigma_2}{\beta_2} \| \mathbf{a} \|^2 - \frac{\sigma_3}{\beta_3} \| \mathbf{b} \|^2 + \frac{\eta_3}{\beta_1} \end{aligned} \tag{20}$$

where η_3 is a positive scalar constant.

Let $\mathbf{z} = [\| \mathbf{e} \| \;\; \| \dot{\mathbf{e}} \| \;\; \| \mathbf{w} \|]^T$, $\mathbf{\Psi} = [\| \mathbf{\Phi}_f \| \;\; \| \mathbf{a} \| \;\; \| \mathbf{b} \|]^T$, $\beta_{min} = \min(\beta_j)$, $\beta_{max} = \max(\beta_j)$, and

$$Q = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} & -\frac{k_2}{2k_1\gamma}(k_{cc}v_{max} + k_{cd}) & 0 \\ -\frac{k_2}{2k_1\gamma}(k_{cc}v_{max} + k_{cd}) & (\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_m(H^*) - k_{cd} & -\lambda_M(H^*) - \frac{k_{cd}}{2\gamma} - \frac{k_{cc}v_{max}}{2\gamma} \\ 0 & -\lambda_M(H^*) - \frac{k_{cd}}{2\gamma} - \frac{k_{cc}v_{max}}{2\gamma} & k_1\gamma \end{bmatrix}$$

and note that Q is positive-definite if γ is chosen large enough. If β_{max}/β_{min} is fixed then there exist positive scalar constants η_4, η_5 that do not increase as γ and β_{min} increase, and positive scalar constants λ_i independent of γ and β_{min}, such that V_3 and $\dot{V}_3$ in (18) and (20) can be bounded as

$$\lambda_1 \| \mathbf{z} \|^2 + \frac{\lambda_2}{\beta_{min}} \| \mathbf{\Psi} \|^2 \leq V_3 \leq \lambda_3 \| \mathbf{z} \|^2 + \frac{\lambda_4}{\beta_{min}} \| \mathbf{\Psi} \|^2$$

$$\dot{V}_3 \leq -(\lambda_m(Q) - \frac{\eta_4}{\gamma} {V_3}^{1/2}) \| \mathbf{z} \|^2 - \frac{\lambda_5}{\beta_{min}} \| \mathbf{\Psi} \|^2 + \frac{\eta_5}{\beta_{min}}$$

Now choose two scalar constants V_M, V_m so that $V_M > V_m \geq V_3(0)$, and define $c_M = \lambda_m(Q) - \eta_4 {V_M}^{1/2}/\gamma$; then choose γ large enough so that $c_M > 0$ (this is always possible). Let $\delta = \max(\lambda_3/c_M, \lambda_4/\lambda_5)$ and choose β_0 so that $\eta_5\delta/\beta_0 < V_m$ (this is always possible). Then selecting $\beta_{min} \geq \beta_0$ ensures that if $V_m \leq V_3 \leq V_M$ then $\dot{V}_3 < 0$. This condition together with $V_M > V_m \geq V_3(0)$ implies that $V_3(t) \leq V_M \;\; \forall t$, so that $c(t) = \lambda_m(Q) - \eta_4 {V_3}^{1/2}(t)/\gamma > c_M > 0 \;\; \forall t$ and

$$\dot{V}_3 \leq -c_M \| \mathbf{z} \|^2 - \frac{\lambda_5}{(\beta_0 + \Delta\beta)} \| \mathbf{\Psi} \|^2 + \frac{\eta_5}{(\beta_0 + \Delta\beta)}$$

where $\Delta\beta = \beta_{min} - \beta_0$ and it is assumed that β_{min} is chosen so that $\Delta\beta > 0$. An analysis identical to the one used in the proof of Theorem 1 now applies and permits the conclusion that $\| \mathbf{z} \|, \| \boldsymbol{\Psi} \|$ are uniformly bounded, which implies that $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$, $\mathbf{f}$, $\mathbf{a}$, and $\mathbf{b}$ are uniformly bounded. Moreover, $\| \mathbf{z} \|, \| \boldsymbol{\Psi} \|$ converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where

$$r_1 = \left(\frac{\delta \eta_5}{\lambda_1(\beta_0 + \Delta\beta)} \right)^{1/2}$$

$$r_2 = \left(\frac{\delta \eta_5}{\lambda_2} \right)^{1/2}$$

Observe that the radius of the ball to which $\| \mathbf{z} \|^2 = \| \mathbf{e} \|^2 + \| \dot{\mathbf{e}} \|^2 + \| \mathbf{w} \|^2$ is guaranteed to converge can be decreased as desired simply by increasing $\Delta\beta$. ■

Several observations can be made concerning the proposed strategy for equilibrium point stabilization of differentially flat nonholonomic mechanical systems. First, this scheme provides a simple and computationally efficient means of ensuring that the system (1) evolves from any initial configuration to any desired goal configuration $\mathbf{x}_d$. The simplicity and efficiency is a consequence of that of the trajectory generation algorithm for the flat outputs and the adaptive tracking controller for the reducing outputs. Trajectory generation can be achieved using standard curve fitting techniques once the flat outputs are identified (see, for example, [7] for a discussion of this problem), and the tracking scheme (15),(16) possesses a simple decentralized structure, utilizes only measurements of system configuration, and requires virtually no information concerning the system dynamics. Thus the proposed control scheme provides an efficient, modular, and readily implementable approach to equilibrium point stabilization. It is shown that the tracking controller ensures uniform boundedness of signals and that the ultimate bounds on the trajectory tracking errors can be reduced arbitrarily by increasing the adaptation gains β_j; notice that exponential convergence ensures that the transient behavior of the system will be well-behaved. It is mentioned that, for clarity of exposition, the calculations required in the proof of Theorem 2 have been condensed significantly in the above presentation; these details are given in the report [29]. An additional observation is that the scalar controller gains k_1, k_2, γ, and β can be replaced with the appropriate diagonal matrices for increased flexibility and performance; scalar gains are used above for simplicity of development. Finally, it is easy to see that the solution to the equilibrium point stabilization problem given in Algorithm 1 also provides a solution to a broad class of trajectory tracking problems for flat systems: given any feasible trajectory for the system the corresponding flat output trajectory $\mathbf{y}_d(t)$ can be determined, and this trajectory can then be tracked using (steps 2 and 3 of) Algorithm 1.

4.3 Implementation Results

The decentralized adaptive equilibrium point stabilization scheme described above is now applied through computer simulation to the two wheel mobile robot shown in Figure 1 and described in (13). This system is differentially flat with flat outputs $\mathbf{y} = [x \quad y]^T$ and reducing outputs $\mathbf{x}_2 = [y \quad \theta]^T$. The simulation study consists of two parts: a system stabilization study and a trajectory tracking investigation. The first simulation illustrates the capability of Algorithm 1 to stabilize the system to the origin from various initial configurations in the presence of external disturbances and with no model information or rate measurements. The trajectory generation phase of Algorithm 1 is accomplished by fitting a fourth order polynomial for each flat output coordinate to the given initial and goal configurations (in all cases the system is initially at rest). This output curve is then used to specify a desired trajectory for the reducing output $\mathbf{x}_{2d}(t)$, and the adaptive tracking scheme (15),(16) is used to track this trajectory. The stabilization algorithm is applied to the mathematical model of the mobile robot described above and shown in Figure 1 through computer simulation with a sampling period of two milliseconds. All integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of two milliseconds. Additional details concerning the simulation strategy for this study can be found in [29]. The system model parameters and disturbances are defined as in the simulation described in Section 3.2, and the adaptive controller parameters k_1, k_2, γ are also set to the values used in that simulation. The adaptive gains $\mathbf{f}$, $\mathbf{a}$, $\mathbf{b}$ are set to zero initially and the adaptation parameters are set as follows: $\sigma_j = 0.1$, $\beta_j = 10$ for $i = 1, 2, 3$. Note that no attempt was made to "tune" the gains to obtain the best possible performance. The stabilization scheme given in Algorithm 1 was tested using a wide range of initial conditions; sample results are given in Figure 3, and indicate that accurate equilibrium point stabilization of the system is achieved.

The next simulation demonstrates the capability of Algorithm 1 to provide accurate trajectory tracking control of the mobile robot shown in Figure 1 and described in (13). Additionally, this simulation indicates that a simple collision avoidance capability can be naturally included with this approach to mobile robot control. The nominal task in this simulation is to track a straight line trajectory in flat output space from $\mathbf{y}_{initial} = [0 \quad 3]^T$ to $\mathbf{y}_{final} = [8 \quad 3]^T$. However, two large rectangular obstacles are placed in the robot workspace as shown in Figure 4, thus preventing completion of the nominal task. In order to provide the robot with a simple collision avoidance capability, Algorithm 1 is modified as follows: if an obstacle is encountered during the task, the nominal path is altered in such a way that the robot tracks along the obstacle until a configuration is reached which permits a smooth output trajectory to connect this configuration to the original goal configuration (see Figure 4). The collision avoidance strategy is essentially "local" in character and provides a setting within which both model-based and sensor-based strategies can be implemented. The scheme is based on the robotic manipulator collision avoidance work reported in [32], and is described in more detail in [29]. The focus here is on the role of Algorithm 1 as a trajectory tracking controller for flat nonholonomic systems; a detailed discussion of the collision avoidance method is beyond the scope of the paper and thus is not provided. Algorithm 1 is implemented exactly as in the first simulation in this section, with the exception of the collision avoidance modification described above. Sample results are given in Figure 4 and indicate that accurate trajectory tracking (and successful collision avoidance) is achieved with the proposed strategy.

5. Adaptive Control of Caplygin Systems

While differential flatness appears to be a useful system property to exploit when solving the trajectory generation problem for (flat) nonholonomic systems, there are at present no general methods for determining whether or not a given system is flat or, if a system is flat, what the flat outputs might be. Thus it is desirable to identify other structural properties of nonholonomic systems that can be used for control purposes, and to develop strategies for controlling systems which possess this structure. In this section we focus our attention on so-called *Caplygin systems* and present an algorithm for stabilizing these systems to an equilibrium configuration in the presence of uncertainty.

5.1 Background

Consider the system (1) with configuration coordinates $\mathbf{x} = [\mathbf{x}_1^T \ \ \mathbf{x}_2^T]^T$, and suppose that the Lagrangian (of the free system) $L(\mathbf{x}, \dot{\mathbf{x}}) = \dot{\mathbf{x}}^T H(\mathbf{x})\dot{\mathbf{x}}/2 - U(\mathbf{x})$ and the input and constraint matrices M, A are independent of $\mathbf{x}_1$. In this case the nonholonomic system (1) becomes

$$\begin{aligned} M(\mathbf{x}_2)\mathbf{T} &= H(\mathbf{x}_2)\ddot{\mathbf{x}} + V_{cc}(\mathbf{x}_2, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}(\mathbf{x}_2) + A^T(\mathbf{x}_2)\lambda \\ A(\mathbf{x}_2)\dot{\mathbf{x}} &= \mathbf{0} \end{aligned} \tag{21}$$

and is known as a (controlled) Caplygin system [12]. This functional form for the system dynamics is an expression of symmetries in the problem, and such symmetries occur naturally in many physical systems of interest in applications [12]. For the Caplygin system (21), the reduced system (3) takes the form

$$\mathbf{F} = H^*(\mathbf{x}_2)\ddot{\mathbf{x}}_2 + V_{cc}^*(\mathbf{x}_2, \dot{\mathbf{x}}_2)\dot{\mathbf{x}}_2 + \mathbf{G}^*(\mathbf{x}_2) \tag{22a}$$

$$\dot{\mathbf{x}}_1 = A^*(\mathbf{x}_2)\dot{\mathbf{x}}_2 \tag{22b}$$

Observe that (22a) is a $2r$ order differential equation which defines the evolution of the $2r$ states $(\mathbf{x}_2, \dot{\mathbf{x}}_2)$. As a consequence, $\mathbf{x}_2$ constitutes a vector of coordinates for the reduced configuration space for the system (22a); this space is often referred to as the "base space" of the complete system (22) [12,17].

The evolution of the base space coordinates can be controlled by properly specifying the input $\mathbf{F}$. Indeed, it is easy to see that the adaptive scheme (15),(16) can be used to cause $\mathbf{x}_2$ to track any desired trajectory $\mathbf{x}_{2d}$ in the presence of model and measurement uncertainty. Given a trajectory for $\mathbf{x}_2$, the behavior of the remaining configuration coordinates $\mathbf{x}_1$ is then completely determined by the kinematic relationship (22b). This observation is used in [12] to develop an algorithm for stabilizing the full system configuration $\mathbf{x}$. The algorithm proposed in [12] is based on the use of geometric phase, which is the extent to which a closed path in the base space fails to be closed in the configuration space, to maneuver both $\mathbf{x}_1$ and $\mathbf{x}_2$ to their desired values. In what follows we present a version of this control algorithm which is implementable without model information or rate measurements.

5.2 Control Scheme

Consider the problem of transferring the Caplygin system (22) from any initial state to an equilibrium point, and suppose for simplicity of notation that the goal configuration is the origin. Let $(\mathbf{x}_1^0, \mathbf{x}_2^0, \dot{\mathbf{x}}_2^0)$ denote the initial state, and consider the following algorithm for stabilizing (22) to the origin.

Algorithm 2:

1. Given an initial state $(\mathbf{x}_1^0, \mathbf{x}_2^0, \dot{\mathbf{x}}_2^0)$, drive the system (22a) to the origin in the $(\mathbf{x}_2, \dot{\mathbf{x}}_2)$ base state space in finite time using the trajectory tracking scheme (15),(16). Observe that this can be accomplished using any smooth path between $(\mathbf{x}_2^0, \dot{\mathbf{x}}_2^0)$ and $(\mathbf{0}, \mathbf{0})$ and that, in general, the resulting state of the system will be $(\mathbf{x}_1^1, \mathbf{0}, \mathbf{0})$ for some nonzero $\mathbf{x}_1^1$.

2. Find a closed path (or series of closed paths) in the base space which produces the desired geometric phase in the configuration space, so that $(\mathbf{x}_1^1, \mathbf{0}, \mathbf{0})$ is transfered to $(\mathbf{0}, \mathbf{0}, \mathbf{0})$.

3. Track the desired base space trajectory $\mathbf{x}_{2d}(t)$ using the controller (15),(16).

In order to successfully implement this approach to stabilizing (22), we must determine a closed path $\mathcal{P}$ in the base space which satisfies the geometric phase condition

$$\mathbf{x}_1^1 = -\int_{\mathcal{P}} A^*(\mathbf{x}_2) d\mathbf{x}_2 \tag{23}$$

Explicit characterization of the closed path $\mathcal{P}$ which satisfies (23) can be achieved for several systems of practical interest; see, for instance, [12,17] and the simulations presented below. However, some systems may require a more general computational approach. For example, an algorithm for approximately characterizing the closed path based on Lie algebraic methods is given in [33]. Alternatively, numerical integration can be utilized to iteratively determine a closed path with the requisite geometric phase [17]. In any case, once this closed path in the base space is found it is clear that Algorithm 2 will produce the desired stabilization of (22).

It is interesting to note that Algorithm 2 is actually applicable to a class of nonholonomic systems which is more general than the class of Caplygin systems (21). To see this, observe that this approach to system stabilization can be implemented with the nonholonomic system

$$\mathbf{F} = H^*(\mathbf{x})\ddot{\mathbf{x}}_2 + V_{cc}^*(\mathbf{x}, \dot{\mathbf{x}}_2)\dot{\mathbf{x}}_2 + \mathbf{G}^*(\mathbf{x}) \tag{24a}$$

$$\dot{\mathbf{x}}_1 = A^*(\mathbf{x}_2)\dot{\mathbf{x}}_2 \tag{24b}$$

in which only the constraint matrix A is independent of $\mathbf{x}_1$ and the Lagrangian and input matrix are functions of the full configuration $\mathbf{x}$. This fact follows directly from the proof of Theorem 2, where it is shown that the tracking scheme (15),(16) is appropriate for the system (24a), and the fact that the evolution of $\mathbf{x}_1$ is still completely determined by the kinematic relationship (24b).

5.3 Implementation Results

The adaptive equilibrium point stabilization scheme described in Algorithm 2 is now applied through computer simulation to two different nonholonomic mechanical systems: the two wheel mobile robot shown in Figure 1 and discussed above, and the "planar skater" depicted in Figure 5 and described in [34]. The first simulation involves the mobile robot. Observe that the change of coordinates $z_1 = x\cos(\theta + \pi/2) + y\sin(\theta + \pi/2)$, $z_2 = \theta + \pi/2$, $z_3 = -x\sin(\theta + \pi/2) + y\cos(\theta + \pi/2)$ permits the model (13) to be rewritten [12]

$$m\ddot{z}_1 = -mz_1\dot{z}_2^2 + p_1(T_1 + T_2) + p_3 z_3(T_1 - T_2) + d_1(t) \tag{25a}$$

$$J\ddot{z}_2 = p_2(T_1 - T_2) + d_2(t) \tag{25b}$$

$$\dot{z}_3 = -z_1\dot{z}_2 \tag{25c}$$

where the p_i are positive constants and all other terms are as defined in (13). It can be seen that this system is Caplygin with base coordinates (z_1, z_2). The dynamics for the base space is defined by (25a),(25b) and the evolution of z_3 is determined through the simple kinematic relation (25c). Note that the kinematics is simple enough to be integrated in closed form in this case, and that the resulting geometric phase condition $z_3^1 = z_1 z_2$ makes the problem of finding a closed path $\mathcal{P}$ in base coordinates which stabilizes z_3 particularly easy.

The control objective in this simulation is to stabilize the system to the origin in the presence of external disturbances and with no model information or rate measurements. This objective is achieved using the three step procedure given in Algorithm 2. In the first step, the initial base configuration (z_1^0, z_2^0) is driven to $(0,0)$ along a straightline path using the tracking controller (15),(16). Then, given the resulting value for z_3^1, a rectangular closed path $\mathcal{P}$ of proper width and length is defined in the base space; this path is then tracked with the adaptive controller (15),(16). The stabilization algorithm is applied to the mathematical model of the mobile robot described above and shown in Figure 1 through computer simulation with a sampling period of two milliseconds. All integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of two milliseconds. Additional details concerning the simulation strategy for this study can be found in [29]. The system model parameters and disturbances are defined as in the simulation presented in Section 3.2, and the adaptive controller (15),(16) is implemented exactly as described in Section 4.3. The stabilization scheme given in Algorithm 2 was tested using a wide range of initial conditions and external disturbances. Sample results are given in Figure 6a-6d, and indicate that accurate equilibrium point stabilization of the mobile robot is achieved.

The next simulation demonstrates the capability of Algorithm 2 to stabilize the orientation of the planar skater shown in Figure 5 and described in [34]. This system represents a simple model of mechanical systems for which angular momentum conservation induces a nonholonomic constraint on the system dynamics. Let the configuration of the system be defined by $\mathbf{x} = [\theta \quad \phi_1 \quad \phi_2]^T$, where θ is the absolute angle of the center body with respect to some fixed coordinate system and ϕ_1, ϕ_2 are the angles of the two arms of the system relative to the center body. The dynamic model for this system is presented in detail in [34], and therefore the specifics are not repeated here. Instead, we simply note that the system is Caplygin, has base coordinates $\mathbf{x}_2 = [\phi_1 \quad \phi_2]^T$ which evolve according to a dynamic model of the general form (22a), and is nonholonomically constrained through conservation of angular momentum. If it assumed that the system has zero momentum initially then this constraint defines the evolution of θ through a kinematic equation of the form (22b). Thus Algorithm 2 is directly applicable to stabilizing the system to an equilibrium configuration.

The control objective in this simulation is to stabilize the planar skater to the origin in configuration space in the presence of external disturbances and with no model information or rate measurements. This objective is achieved using the three step procedure given in Algorithm 2. This stabilization algorithm is implemented exactly as it was with the mobile robot in the first part of this section, with the exception of the determination of a closed path $\mathcal{P}$ in base coordinates which stabilizes θ. In the present case the geometric phase condition (23) could not be integrated in closed form, so that the calculation of an appropriate path $\mathcal{P}$ given the value $x_1^1 = \theta^1$ was performed numerically. The stabilization scheme was tested using a wide range of initial conditions. Sample results are given in Figure 7a-7f, and indicate that accurate equilibrium point stabilization of the planar skater is achieved. Figure 7a depicts the initial configuration of the skater and Figure 7b shows the (user defined) "zero" configuration for the base coordinates; observe that in driving the system to the origin in base space the θ coordinate does not reach zero. Figures 7c-7f then shows a closed path in base space which causes the θ coordinate to be transfered from its value in Figure 7b (approximately 45°) to its desired final value of 0°.

6. Conclusions

This paper considers the problem of controlling nonholonomic mechanical systems in the presence of incomplete information concerning the system model and state, and presents a new class of decentralized adaptive controllers as a solution to this problem. The proposed control strategies provide simple and robust solutions to a number of important nonholonomic system control problems, including stabilization of general systems to an equilibrium manifold and stabilization of differentially flat and Caplygin systems to an equilibrium point. Additionally, it is indicated how the proposed approach to equilibrium point stabilization of flat systems can be applied to trajectory tracking control. All of the schemes possess a simple decentralized structure, are computationally efficient and implementable without system model or rate information, and ensure uniform boundedness of all signals and accurate motion control. Computer simulation results are provided which demonstrate stabilization of a mobile robot, stabilization of a planar skater, and trajectory tracking control with simple collision avoidance for a mobile robot. Future work will focus on the development of adaptive equilibrium point stabilization schemes for additional classes of nonholonomic mechanical systems.

7. Acknowledgments

The research described in this paper was supported in part through contracts with the Army Research Office and Sandia National Laboratories.

8. References

1. Li, Z. and J. Canny, *Nonholonomic Motion Planning*, Kluwer, Norwell, MA, 1993

2. Astolfi, A., *Asymptotic Stabilization of Nonholonomic Systems with Discontinuous Control*, Doctoral dissertation, Swiss Federal Institute of Technology, 1995

3. Sordalen, O. and O. Egeland, "Exponential Stabilization of Nonholonomic Chained Systems", *IEEE Transactions on Automatic Control*, Vol. 40, No. 1, 1995, pp. 35-49

4. Tilbury, D., R. Murray, and S. Sastry, "Trajectory Generation for the N-Trailer Problem Using Goursat Normal Form", *IEEE Transactions on Automatic Control*, Vol. 40, No. 5, 1995, pp. 802-819

5. Pappas, G. and K. Kyriakopoulo, "Stabilization of Nonholonomic Vehicles Under Kinematic Constraints", *International Journal of Control*, Vol. 61, No. 4, 1995, pp. 933-947

6. Bushnell, L., D. Tilbury, and S. Sastry, "Steering Three-Input Nonholonomic Systems: The Fire Truck Example", *International Journal of Robotics Research*, Vol. 14, No. 4, 1995, pp. 366-381

7. Fliess, M., J. Levine. P. Martin, and P. Rouchon, "Flatness and Defect of Nonlinear Systems: Introductory Theory and Examples", *International Journal of Control*, Vol. 61, No. 6, 1995, pp. 1327-1361

8. Leonard, N. and P. Krishnaprasad, "Motion Control of Drift-Free, Left-Invariant Systems on Lie Groups", *IEEE Transactions on Automatic Control*, Vol. 40, No. 9, 1995, pp. 1539-1554

9. Kelly, S. and R. Murray, "Geometric Phases and Robotic Locomotion", *Journal of Robotic Systems*, Vol. 12, No. 6, 1995, pp. 417-431

10. Grabbe, M., J. Carroll, D. Dawson, and Z. Qu, "Review and Unification of Reduced-Order Force Control Methods", *Journal of Robotic Systems*, Vol. 10, No. 4, 1993, pp. 481-504

11. Bloch, A., P. Krishnaprasad, J. Marsden, and R. Murray, "Nonholonomic Mechanical Systems with Symmetry", Technical Report CIT/CDS 94-013, California Institute of Technology, 1994

12. Bloch, A., M. Reyhanoglu, and N. McClamroch, "Control and Stabilization of Nonholonomic Dynamic Systems", *IEEE Transactions on Automatic Control*, Vol. 37, No. 11, 1992, pp. 1746-1757

13. Sarkar, N., X. Yun, and V. Kumar, "Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile Robots", *International Journal of Robotics Research*, Vol. 13, No. 1, 1994, pp. 55-69

14. Su, C. and Y. Stepanenko, "Robust Motion/Force Control of Mechanical Systems with Classical Nonholonomic Constraints", *IEEE Transactions on Automatic Control*, Vol. 39, No. 3, 1994, pp. 609-614

15. M'Closkey, R. and R. Murray, "Extending Exponential Stabilizers for Nonholonomic Systems from Kinematic Controllers to Dynamic Controllers", *Proc. IFAC Symposium on Robot Control*, Capri, Italy, September 1994

16. Maschke, B. and A. van der Schaft, "A Hamiltonian Approach to Stabilization of Nonholonomic Mechanical Systems", *Proc. 33rd IEEE Conference on Decision and Control*, Orlando, FL, December 1994

17. Murray, R., "Nonlinear Control of Mechanical Systems: A Lagrangian Perspective", *Proc. IFAC Symposium on Nonlinear Control System Design*, Lake Tahoe, CA, June 1995 1995

18. Craig, J., P. Hsu, and S. Sastry, "Adaptive Control of Mechanical Manipulators", *International Journal of Robotics Research*, Vol. 6, No. 2, 1987, pp. 16-28

19. Slotine, J-J. and W. Li, "On the Adaptive Control of Robot Manipulators", *International Journal of Robotics Research*, Vol. 6, No. 3, 1987, pp. 49-59

20. Bayard, D. and J. Wen, "New Class of Control Laws for Robotic Manipulators: Adaptive Case", *International Journal of Control*, Vol. 47, No. 5, 1988, pp. 1387-1406

21. Ortega, R. and M. Spong, "Adaptive Motion Control of Rigid Robots: A Tutorial", *Automatica*, Vol. 25, No. 6, 1989, pp. 877-888

22. Colbaugh, R., K. Glass, and H. Seraji, "Performance-Based Adaptive Tracking Control of Robot Manipulators", *Journal of Robotic Systems*, Vol. 12, No. 8, 1995, pp. 517-530

23. Colbaugh, R., H. Seraji, and K. Glass, "Adaptive Compliant Motion Control for Dexterous Manipulators", *International Journal of Robotics Research*, Vol. 14, No. 3, 1995, pp. 270-280

24. Colbaugh, R., K. Glass, and E. Barany, "Adaptive Stabilization and Tracking Control of Electrically-Driven Manipulators", *Journal of Robotic Systems*, Vol. 13, No. 4, 1996 (in press)

25. Seraji, H. and R. Colbaugh, "Force Tracking in Impedance Control", *International Journal of Robotics Research*, Vol. 15, 1996 (in press)

26. Nijmeijer, H. and A. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer-Verlag, New York, 1990

27. Brockett, R., "Asymptotic Stability and Feedback Stabilization", *Differential Geometric Control Theory*, Birkhauser, Boston, 1983

28. Saletan, E. and A. Cromer, *Theoretical Mechanics*, Wiley, New York, 1971

29. Colbaugh, R., "Adaptive Control of Nonholonomic Mechanical Systems", Robotics Laboratory Report 95-10, New Mexico State University, May 1995

30. Ioannou, P., "Robust Adaptive Controller with Zero Residual Tracking Errors", *IEEE Transactions on Automatic Control*, Vol. 31, No. 8, 1986, pp. 773-776

31. Corless, M. and G. Leitmann, "Continuous State Feedback Guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems", *IEEE Transactions on Automatic Control*, Vol. 26, No. 5, 1981, pp. 1139-1144

32. Glass, K., R. Colbaugh, D. Lim, and H. Seraji, "Real-Time Collision Avoidance for Redundant Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, 1995, pp. 448-457

33. Laffarriere, G. and H. Sussmann, "Motion Planning for Completely Controllable Systems without Drift", *Proc. IEEE International Conference on Robotics and Automation*, Sacramento, CA, April 1991

34. Walsh, G. and S. Sastry, "On Reorienting Linked Rigid Bodies Using Internal Motions", *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 1, 1995, pp. 139-146

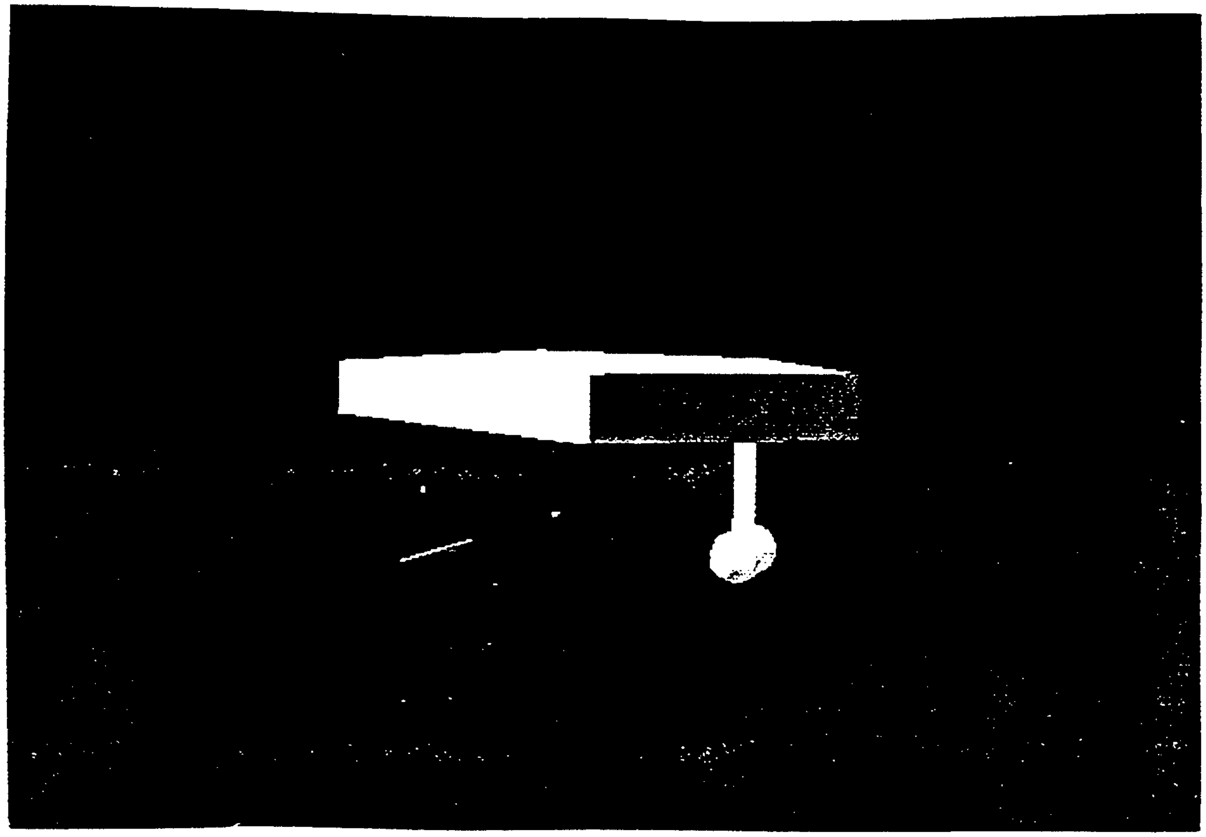


Figure 1: Graphical Representation of Mobile Vehicle

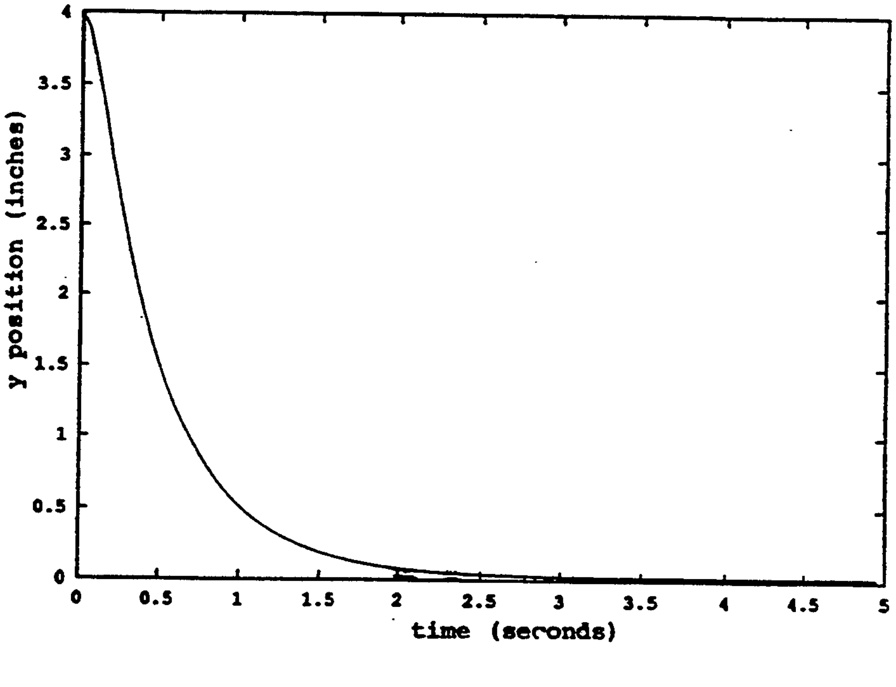


Figure 2a: Response of output coordinate y in equilibrium manifold stabilization simulation

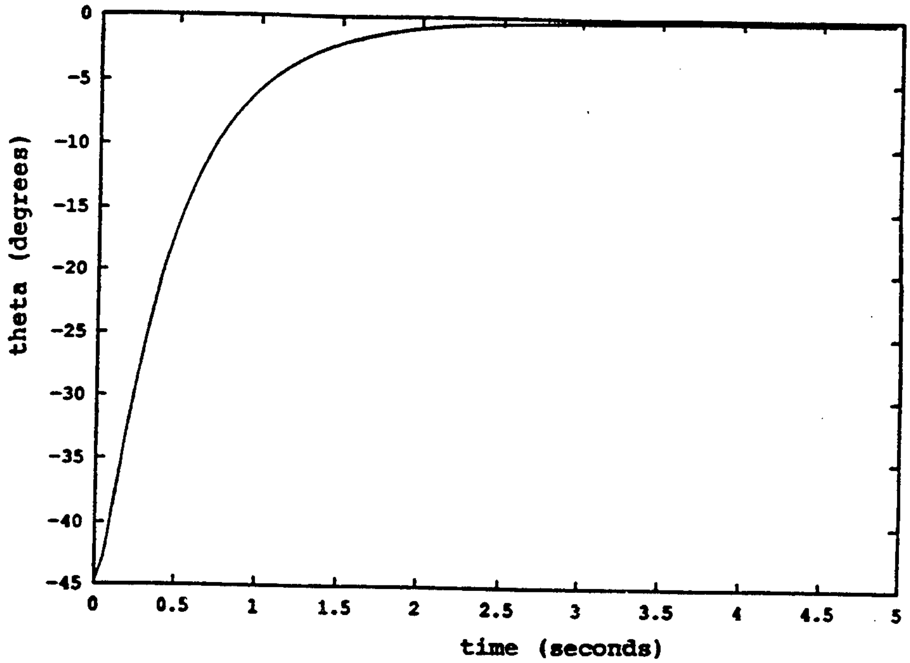


Figure 2b: Response of output coordinate θ in equilibrium manifold stabilization simulation

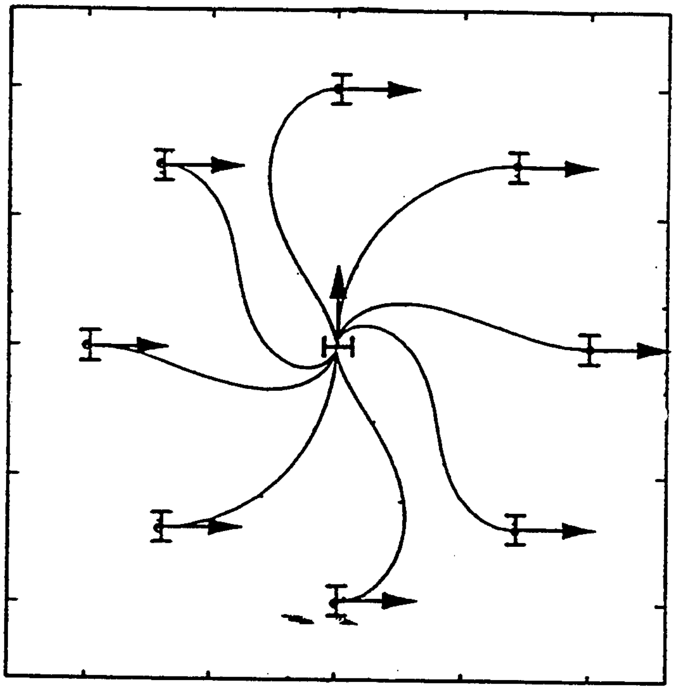


Figure 3: Sample responses in equilibrium point stabilization simulation

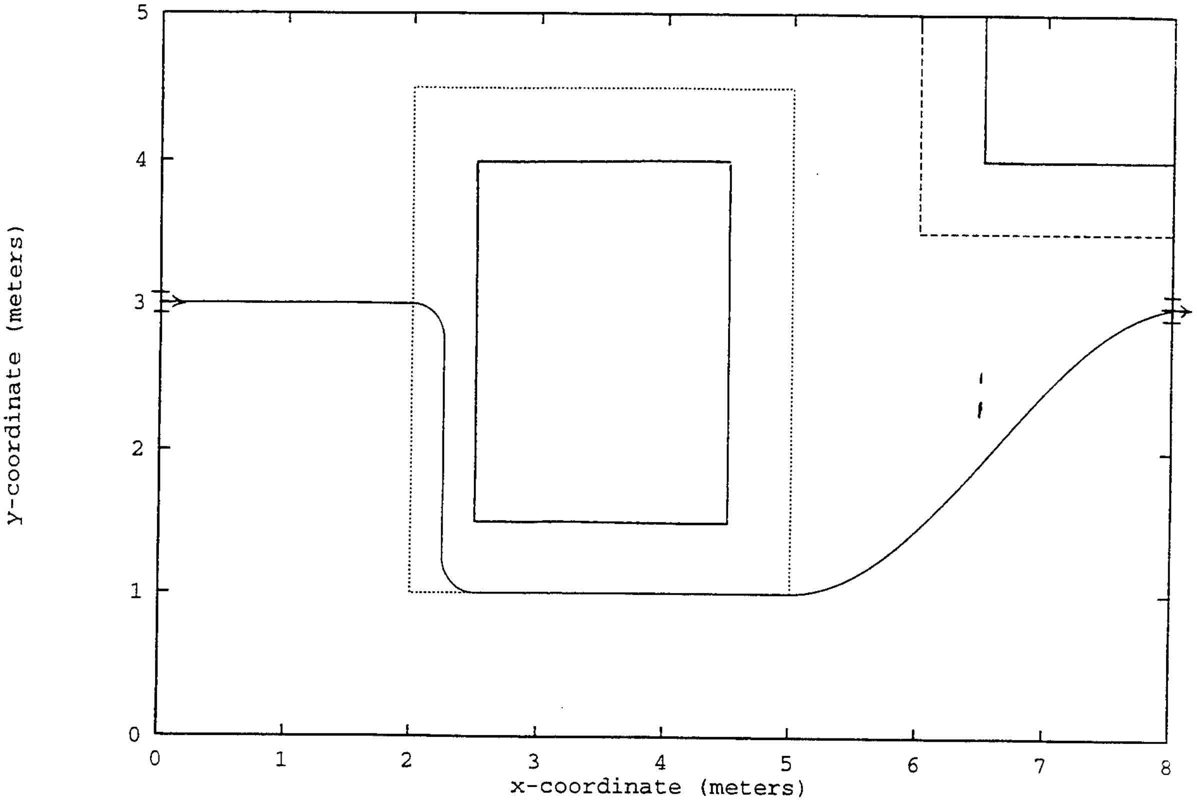


Figure 4: Example of Simple Collision Avoidance

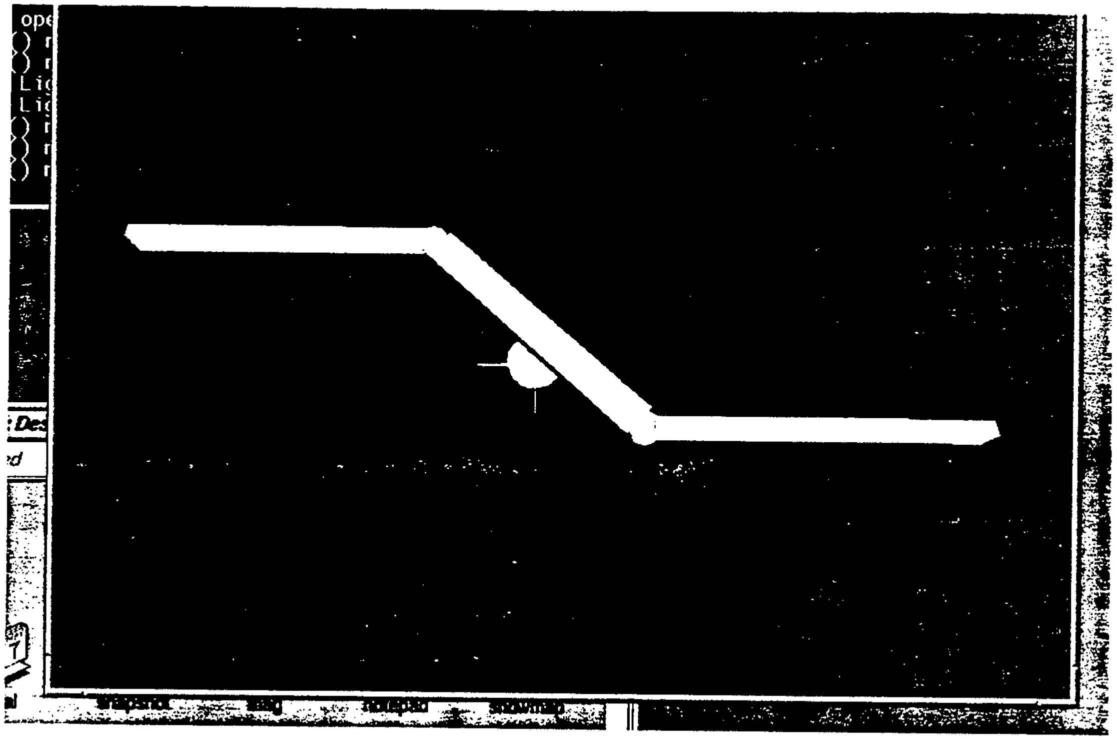


Figure 5: Graphical representation of planar skater

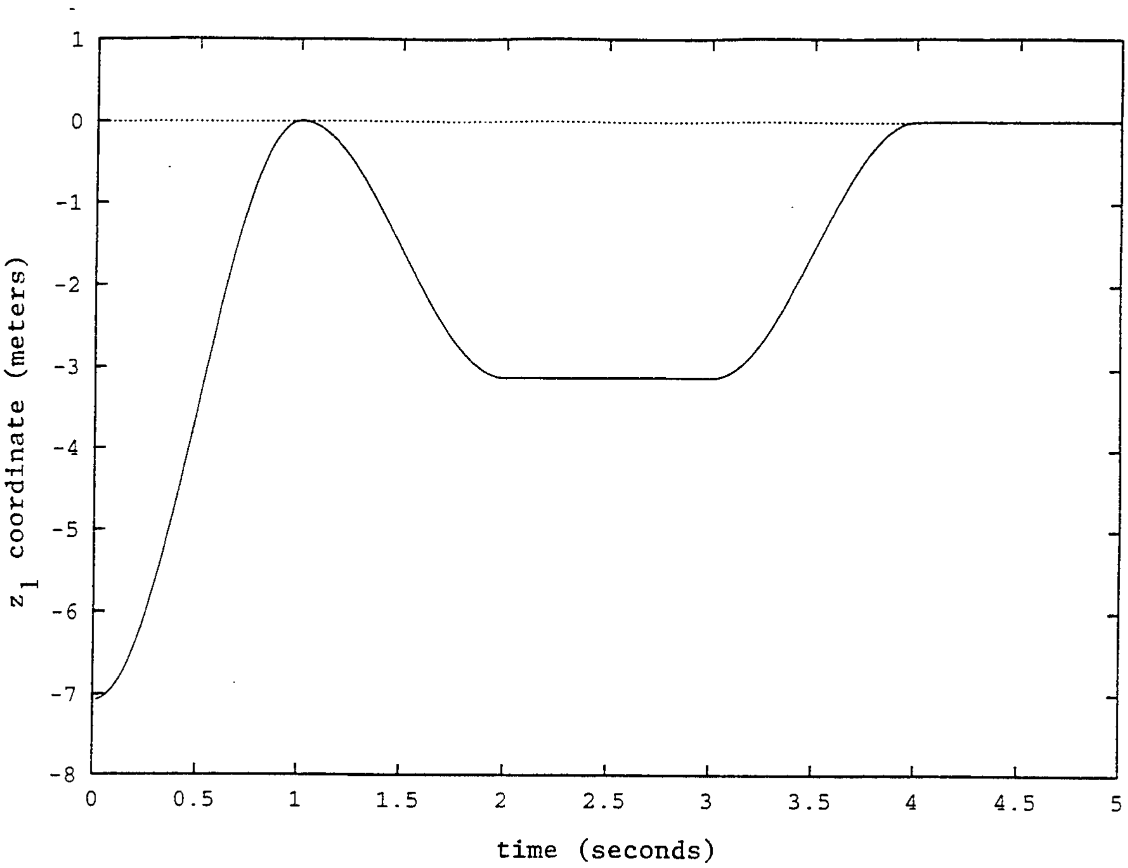


Figure 6a: Response of base coordinate z_1 in mobile robot simulation

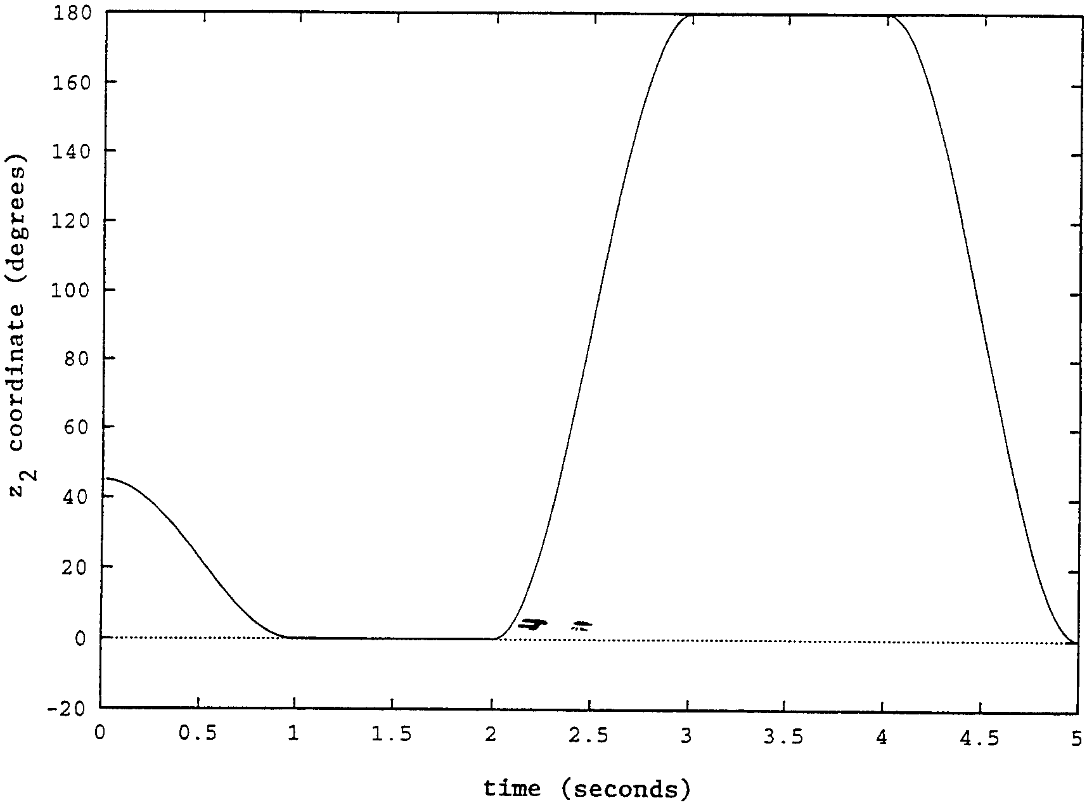


Figure 6b: Response of base coordinate$_2 z_{2_3}$$_9$in mobile robot simulation

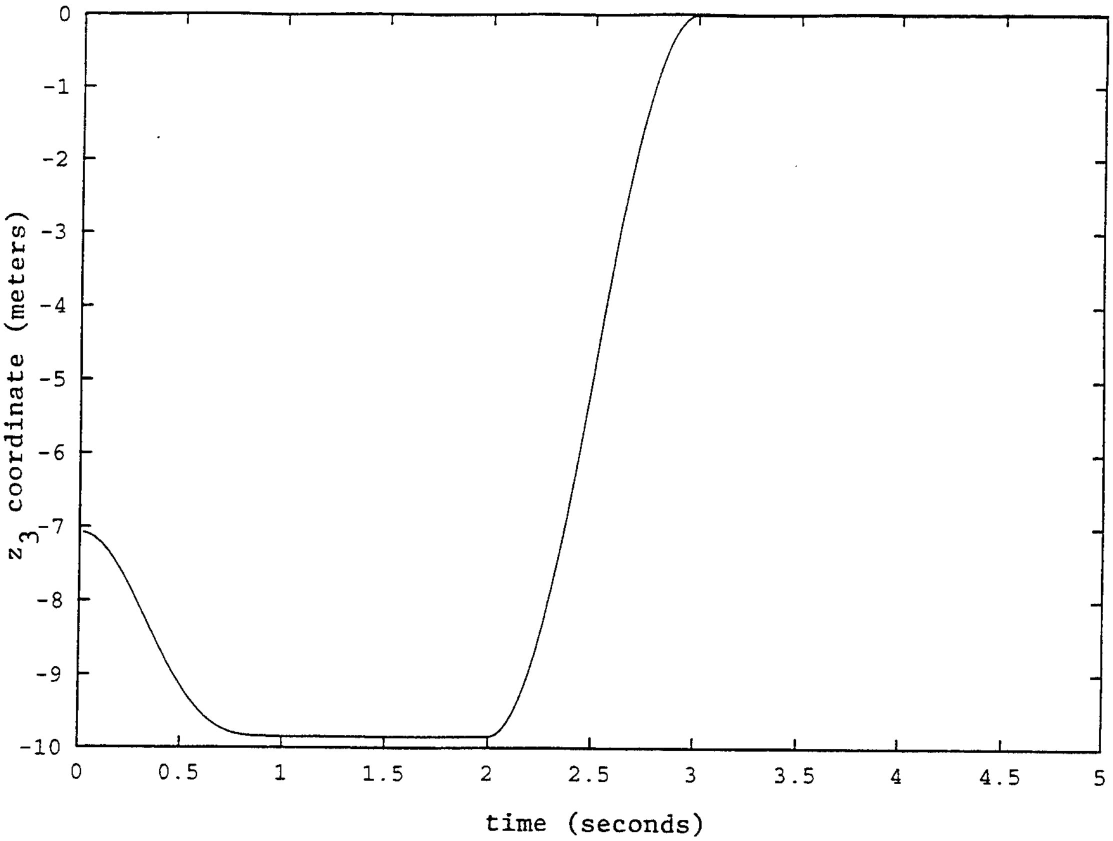


Figure 6c: Response of coordinate z_3 in mobile robot simulation

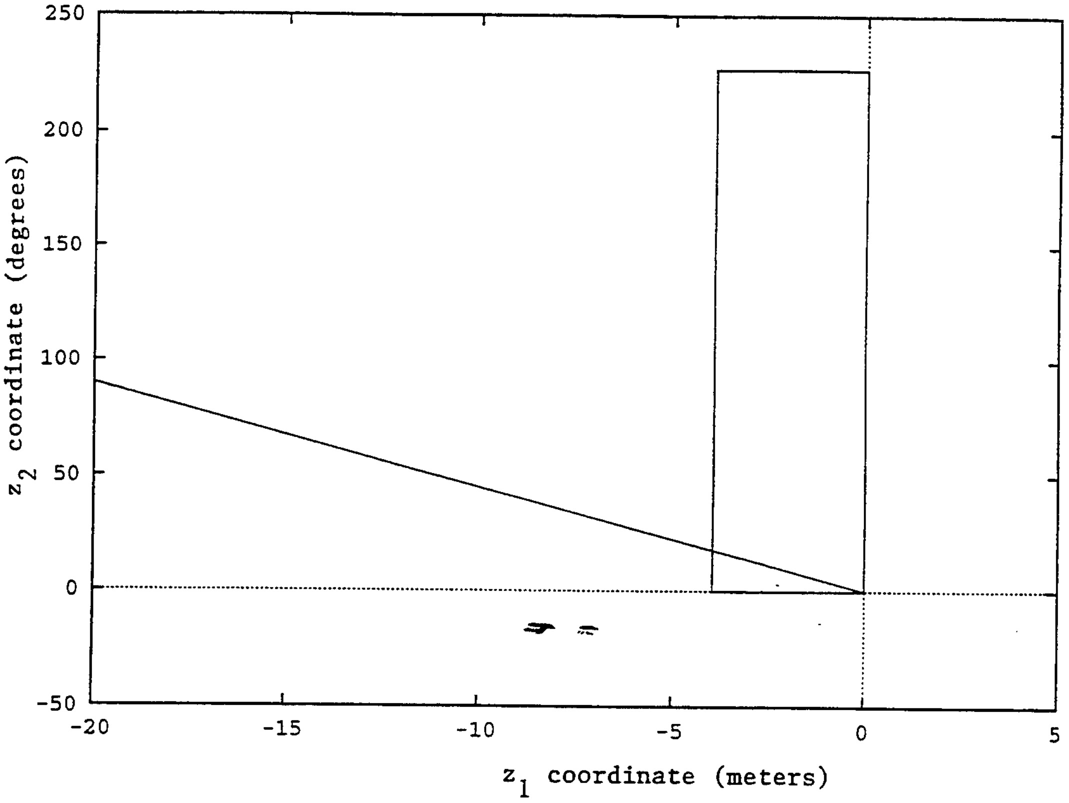
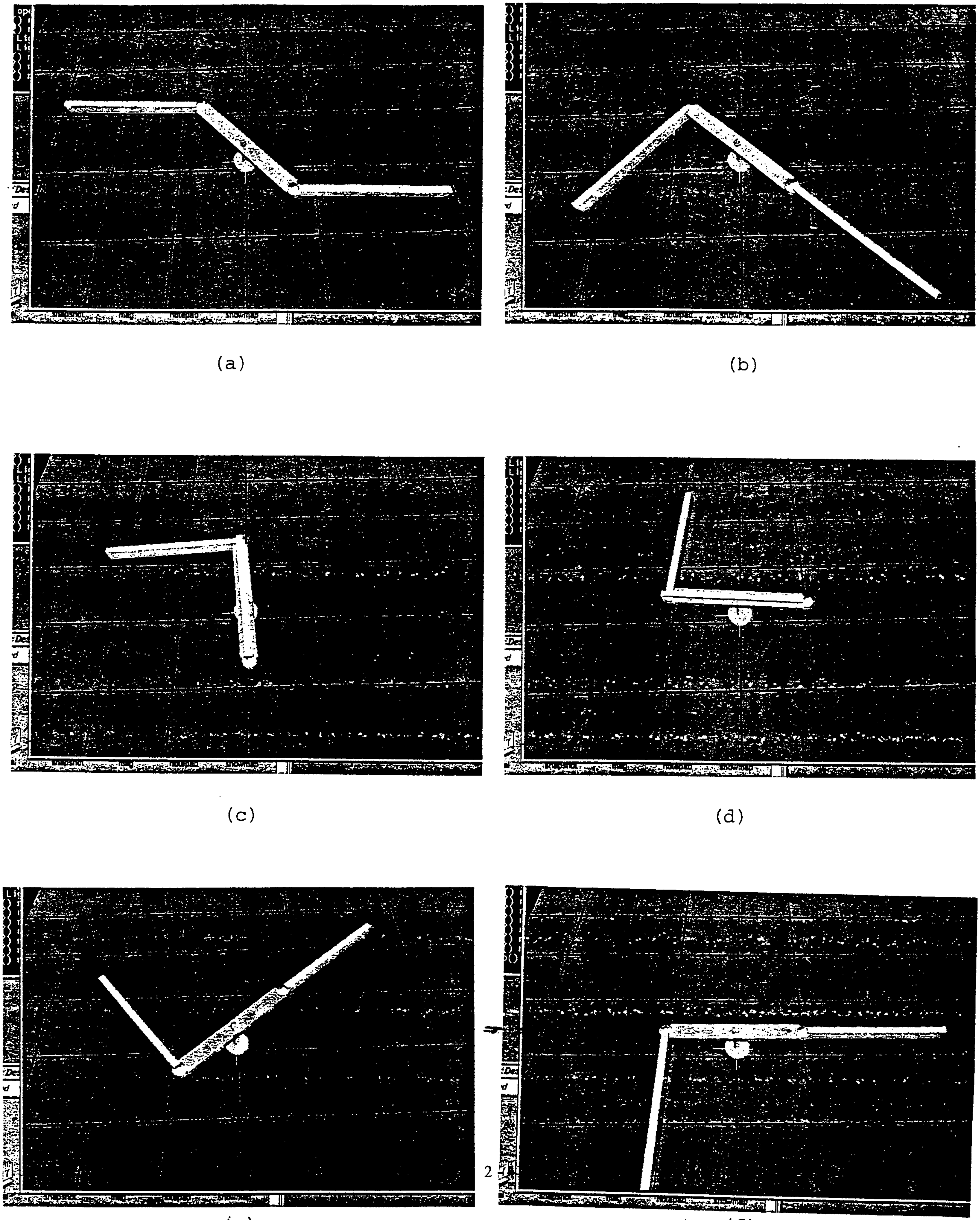


Figure 6d: Closed-path in base-space in mobile robot simulation



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7: Evolution of skater configuration in equilibrium point stabilization simulation

submitted for publication

ADAPTIVE COMPLIANT MOTION CONTROL OF ELECTRICALLY-DRIVEN MANIPULATORS

R. Colbaugh **K. Glass**

Department of Mechanical Engineering
New Mexico State University, Las Cruces, NM 88003 USA

Abstract

This paper presents two adaptive schemes for compliant motion control of uncertain rigid-link, electrically-driven manipulators. The first strategy is developed using an *adaptive impedance control* approach and is appropriate for tasks in which the dynamic character of the end-effector/environment interaction must be controlled, while the second scheme is an *adaptive position/force controller* and is useful for those applications that require independent control of end-effector position and contact force. The proposed controllers are very general and computationally efficient, and can be implemented with virtually no *a priori* information concerning the manipulator/actuator dynamic model or the environment. It is shown that the schemes ensure (semiglobal) uniform boundedness of all signals, and that the ultimate size of the system errors can be made arbitrarily small. The capabilities of the proposed control strategies are illustrated through both computer simulations and laboratory experiments with a six degree-of-freedom (DOF) IMI Zebra Zero manipulator.

1. Introduction

The problem of controlling robotic manipulators that come into contact with objects in their workspace is of central importance in many applications. Manipulators are subject to interaction forces whenever they perform tasks involving motion which is constrained by the environment, such as assembling parts or executing grinding or finishing operations. In such motions, the interaction forces must be accommodated rather than resisted, so as to comply with the environmental constraints. Recognizing the importance of such compliant motion, many researchers have studied this problem for rigid-link manipulators. This intensive investigation has produced two basic approaches to achieving compliant motion: position/force control [e.g., 1] and impedance control [e.g., 2]. The first approach is based on the observation that when the manipulator end-effector is in contact with the environment, the space of the end-effector coordinates can be naturally decomposed into a "position subspace" and a "force subspace"; these two subspaces correspond to the directions in which the end-effector is, respectively, able to move and able to exert forces

2-42

on the environment. The position/force control approach to compliant motion is then to define and track a position (and orientation) trajectory in the position subspace and a force (and moment) trajectory in the force subspace. Alternatively, the impedance control approach proposes that the control objective should be the regulation of the mechanical impedance of the robot end-effector. Thus the objective of the impedance controller is to maintain a desired dynamic relationship between the end-effector position and the end-effector/environment contact force. It is noted that each of the approaches to compliant motion control summarized above has desirable properties and is well-suited to particular robotic tasks. For example, position/force control permits the end-effector position and contact force to be explicitly and independently controlled, which is necessary in applications such as contact-based surface inspection or remediation of hazardous and unstructured environments. Alternatively, the impedance control formulation provides a unified framework for considering both unconstrained and constrained motion control problems and for accommodating transition between these two operating modes; as a consequence impedance control has been found useful for tasks such as deburring of machined parts or mechanical assembly.

During the past decade, there has been considerable interest in designing compliant motion control strategies for rigid-link manipulators which can perform well despite incomplete information concerning the system model. Most of the controllers proposed thusfar have been developed by using either adaptive control methods or robust control techniques to compensate for uncertainty in the manipulator model. It is noted that adaptive control methods in particular have been shown to provide an attractive means of handling the model uncertainty. Included among the advantages of this approach is a reduced dependence upon *a priori* model information, the capacity to respond quickly to unexpected events, and the potential to continuously improve performance while the manipulator is in operation. An important limitation of the controllers summarized above is that these schemes have been designed by neglecting the actuator dynamics and assuming that the joint torques can be commanded directly. However, as demonstrated in [e.g., 3-5], the actuator dynamics constitute an important component of the complete manipulator dynamic model. Recognizing the potential difficulties associated with ignoring the effects of the actuator dynamics, several researchers have recently considered the problem of controlling the motion of rigid-link, electrically-driven (RLED) manipulators [5-16]. In much of this work, the controller development requires full knowledge of the complex dynamic models for the manipulator and actuators [e.g., 5-9]. Research in which controllers are designed with the capability to compensate for uncertainty in the manipulator/actuator system includes work on robust control strategies [10,11] and adaptive schemes [12-16].

It is noted that implementation of most of these robust and adaptive controllers requires the calculation of complex manipulator-specific quantities, such as the time-derivative of the manipulator "regressor" matrix or upper bounds on the derivatives of the "fictitious controls"; this, in turn, limits the generality and applicability of these strategies. Additionally, most of this research has focused on the (unconstrained) motion control problem, despite the fact that the effects of the actuator dynamics is expected to very important in constrained motion applications. Some recent progress on compliant motion control of RLED manipulators is reported in [17].

This paper considers the compliant motion control problem for uncertain RLED manipulators, and presents two new adaptive schemes as solutions to this problem. The proposed controllers possess a simple and modular structure, are easy to implement, and requires virtually no information regarding the manipulator/actuator system model. The first scheme is an impedance controller which adaptively generates the joint torques required to provide the desired dynamic relationship between end-effector position and contact force, while the second strategy is an adaptive position/force controller which ensures that the desired end-effector position trajectory/force setpoint is achieved. It is shown that the schemes ensure semiglobal uniform boundedness of all signals, and that the ultimate size of the system errors can be made arbitrarily small. The efficacy of the proposed schemes is illustrated through computer simulations and hardware experiments with an IMI Zebra Zero manipulator.

2. Preliminaries

This paper considers the adaptive compliant motion control problem for RLED manipulators. Let $\mathbf{p} \in \Re^m$ define the position and orientation of the manipulator end-effector relative to a fixed user-defined reference frame and $\theta \in \Re^n$ denote the vector of manipulator joint coordinates. Then the forward kinematic and differential kinematic maps between the robot joint coordinates θ and the end-effector coordinates $\mathbf{p}$ can be written as

$$\mathbf{p} = \mathbf{h}(\theta), \qquad \dot{\mathbf{p}} = J(\theta)\dot{\theta} \tag{1}$$

where $\mathbf{h} : \Re^n \rightarrow \Re^m$ is smooth and $J \in \Re^{m \times n}$ is the end-effector Jacobian matrix.

Observe that there are numerous advantages to formulating the manipulator control problem directly in terms of the end-effector coordinates $\mathbf{p}$. For example, these coordinates are typically more task-relevant than the joint coordinates θ, so that developing the controller in terms of $\mathbf{p}$ can lead to improved performance, efficiency, and implementability. Note that such a formulation is particularly useful for compliance control applications, since the compliant motion objectives are stated most naturally in terms of task-space variables. If the manipulator is nonredundant (so that $m = n$) and is in a region of the workspace where J has full rank, then $\mathbf{p}$ and θ are diffeomorphic and this formulation presents no difficulties. A task-space formulation can also be realized if the manipulator is kinematically redundant (so that $m < n$) by utilizing, for example, the *configuration control* approach [e.g., 18,19]. In what follows, we shall consider nonredundant and redundant robots together and introduce a set of n task-space coordinates $\mathbf{x}$ obtained by augmenting $\mathbf{p}$ with $n - m$ kinematic functions that define some auxiliary task to be performed by the manipulator [18,19]. To retain generality, we shall require only that the kinematic relationship between θ and $\mathbf{x}$ is known and smooth and can be written in a form analogous to (1):

$$\mathbf{x} = \mathbf{h}_a(\theta), \qquad \dot{\mathbf{x}} = J_a(\theta)\dot{\theta} \tag{2}$$

where $\mathbf{h}_a : \Re^n \rightarrow \Re^n$ and $J_a \in \Re^{n \times n}$. Observe that for $\mathbf{x}$ to be a valid task-space coordinate vector the elements of $\mathbf{x}$ must be independent in the region of interest; thus it will be assumed in our development that J_a is of full rank.

Consider an n DOF rigid-link manipulator actuated by armature-controlled DC motors. Let $\mathbf{T} \in \Re^n$ denote the vector of associated joint torques, $\mathbf{u} \in \Re^n$ represent the vector of armature voltages, and $\mathbf{I} \in \Re^n$ define the vector of armature currents. The dynamic model for this RLED manipulator system is a $3n$th order differential equation relating the joint coordinates θ and the system control input $\mathbf{u}$. For example, in the event that θ, $\dot{\theta}$, and $\mathbf{T}$ are measurable, the following dynamic model for the manipulator/actuator system is convenient for subsequent analysis [e.g., 5]:

$$\mathbf{T} = H_\theta(\theta)\ddot{\theta} + V_\theta(\theta, \dot{\theta})\dot{\theta} + \mathbf{G}_\theta(\theta) + J_a^T(\theta)\mathbf{P} \tag{3a}$$

$$\mathbf{u} = M\dot{\mathbf{T}} + R\mathbf{T} + K\dot{\theta} \tag{3b}$$

where $H_\theta \in \Re^{n \times n}$ is the system inertia matrix, $V_\theta \in \Re^{n \times n}$ and $\mathbf{G}_\theta \in \Re^n$ quantify Coriolis/centripetal acceleration and gravity effects, respectively, $\mathbf{P} \in \Re^n$ is the vector of forces and moments exerted by the end-effector on the environment, and $M, R, K \in \Re^{n \times n}$ are positive, constant, diagonal matrices which characterize the actuator dynamics. Alternatively, if θ, $\dot{\theta}$, and $\mathbf{I}$ are measurable, then the following equivalent representation of the system dynamics is typically more suitable for controller design [5]:

$$K_T\mathbf{I} = H_\theta(\theta)\ddot{\theta} + V_\theta(\theta, \dot{\theta})\dot{\theta} + \mathbf{G}_\theta(\theta) + J_a^T(\theta)\mathbf{P} \tag{4a}$$

$$\mathbf{u} = MK_T\dot{\mathbf{I}} + RK_T\mathbf{I} + K\dot{\theta} \tag{4b}$$

where $K_T \in \Re^{n \times n}$ is the positive, constant, diagonal matrix which relates current and torque via $\mathbf{T} = K_T\mathbf{I}$. Note that, for convenience of development, we shall assume in what follows that the environment is linearly elastic (so that the magnitude of $\mathbf{P}$ is proportional to the environmental deformation), although our results remain valid in more general situations.

It is useful for our subsequent analysis to represent the dynamics (3) in terms of the task-space coordinates $\mathbf{x}$. Developing such a representation is standard [e.g, 20], and the resulting task-space model can be written as

$$\mathbf{F} = H\ddot{\mathbf{x}} + V_{cc}\dot{\mathbf{x}} + \mathbf{G} + \mathbf{P}, \qquad \mathbf{T} = J_a^T\mathbf{F} \tag{5a}$$

$$\mathbf{u} = M\dot{\mathbf{T}} + R\mathbf{T} + K\dot{\theta} \tag{5b}$$

where $\mathbf{F} \in \Re^n$ is the generalized force associated with $\mathbf{x}$ and $H, V_{cc}, \mathbf{G}$ are the task-space counterparts to $H_\theta, V_\theta, \mathbf{G}_\theta$ in (3a). It is well known that the rigid-link manipulator dynamics (5a) possesses considerable structure. For example, for any set of generalized coordinates $\mathbf{x}$, the dynamic model terms $H, \mathbf{G}$ are bounded functions of $\mathbf{x}$ whose time derivatives $\dot{H}, \dot{\mathbf{G}}$ are also bounded in $\mathbf{x}$ and depend linearly on $\dot{\mathbf{x}}$, the matrix H is symmetric and positive-definite, the matrix V_{cc} is bounded in $\mathbf{x}$ and depends linearly on $\dot{\mathbf{x}}$, and the matrices H and V_{cc} are related according to $\dot{H} = V_{cc} + V_{cc}^T$. Additionally, $V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\mathbf{y} = V_{cc}(\mathbf{x}, \mathbf{y})\dot{\mathbf{x}}$ $\forall \mathbf{y}$, and if $\mathbf{y}$ and $\dot{\mathbf{y}}$ are bounded then $V_{cc}(\mathbf{x}, \mathbf{y})$ is bounded and $\dot{V}_{cc}(\mathbf{x}, \mathbf{y})$ grows linearly with $\dot{\mathbf{x}}$ [e.g., 20]. All of these properties will prove useful for controller development.

In this paper we shall address both the impedance control problem, in which the control objective is to regulate the dynamic character of the end-effector/environment interaction, and the position/force control problem, in which explicit and independent control of end-effector position and contact force is desired. In what follows, it is assumed that θ, $\dot{\theta}$, $\mathbf{T}$, and $\mathbf{P}$ are measurable, so that the control strategies will be developed for the dynamical system (5). It should be noted, however, that the design procedure developed here is directly applicable to the task-space version of the dynamical system (4) for the case in which θ, $\dot{\theta}$, $\mathbf{I}$, and $\mathbf{P}$ can be measured.

Observe that the dynamic model (5) consists of two cascaded dynamical systems. One consequence of this structure is that the joint torque $\mathbf{T}$ cannot be commanded directly, as is assumed in the design of controllers at the torque input level, and instead must be realized as the output of the actuator dynamics (5b) through proper specification of the control input $\mathbf{u}$. The structure of the RLED manipulator dynamics (5) suggests partitioning the control system design problem into two subproblems:

manipulator control: regard $\mathbf{T}$ as the control input for the subsystem (5a) and specify the desired evolution of this variable $\mathbf{T}_d(t)$ in such a way that if $\mathbf{T} = \mathbf{T}_d$ then the given control objective would be achieved

actuator control: specify the actual control input $\mathbf{u}$ so that $\mathbf{T}$ closely tracks $\mathbf{T}_d$

This approach to controller design is adopted in this paper, so that each of the proposed control systems consists of two subsystems: an adaptive strategy that provides the (fictitious) control input $\mathbf{T}_d$ required to ensure that the system (5a) evolves as desired, and an adaptive scheme that determines the (actual) control input $\mathbf{u}$ which guarantees that the system (5b) evolves with $\mathbf{T}$ closely tracking $\mathbf{T}_d$. It is stressed that, although each of the subsystems is essentially designed separately, it will be explicitly verified that the subsystems can be combined to yield a complete control system with the desired stability and convergence properties.

Our approach to the design and analysis of suitable controllers is based on Lyapunov stability theory. The following ultimate boundedness lemma summarizes the properties of a certain class of Lyapunov functions and will be of direct relevance in our development.

Lemma UB: Consider the coupled dynamical system $\dot{\mathbf{x}}_1 = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, t)$, $\dot{\mathbf{x}}_2 = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, t)$. Let $V(\mathbf{x}_1, \mathbf{x}_2, t)$ be a Lyapunov function for the system with the properties

$$\lambda_1 \parallel \mathbf{x}_1 \parallel^2 + \lambda_2 \parallel \mathbf{x}_2 \parallel^2 \leq V \leq \lambda_3 \parallel \mathbf{x}_1 \parallel^2 + \lambda_4 \parallel \mathbf{x}_2 \parallel^2$$
$$\dot{V} \leq -\lambda_5 \parallel \mathbf{x}_1 \parallel^2 - \lambda_6 \parallel \mathbf{x}_2 \parallel^2 + \epsilon$$

where ϵ and the λ_i are positive scalar constants. Define $\delta = \max(\lambda_3/\lambda_5, \lambda_4/\lambda_6)$ and $r_i = (\delta\epsilon/\lambda_i)^{1/2}$ for $i = 1, 2$. Then for any initial state $\mathbf{x}_1(0), \mathbf{x}_2(0)$ the system will evolve so that $\mathbf{x}_1(t), \mathbf{x}_2(t)$ are uniformly bounded and converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where $B_{r_i} = \{\mathbf{x}_i \ : \parallel \mathbf{x}_i \parallel \leq r_i\}$.

Proof: Straightforward application of the global exponential convergence theorem of Corless [21] gives the result (see also [22]). ■

3. Adaptive Impedance Controller

In this section we develop an adaptive impedance control system for uncertain RLED manipulators. When the end-effector of the manipulator is in contact with the environment, the objective of impedance control is to cause the end-effector to respond to contact forces according to some user-defined dynamics. For example, the desired dynamic relationship between the end-effector position $\mathbf{x}$, the end-effector reference trajectory $\mathbf{x}_r \in \Re^n$, and the end-effector/environment contact force $\mathbf{P}$ is often specified as follows:

$$M_{imp}(\ddot{\mathbf{x}}_r - \ddot{\mathbf{x}}) + C_{imp}(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}) + K_{imp}(\mathbf{x}_r - \mathbf{x}) = \mathbf{P} \tag{6}$$

where $\mathbf{x}_r$ is bounded with bounded derivatives and the matrices $M_{imp}, C_{imp}, K_{imp} \in \Re^{n \times n}$ are specified by the user so that the dynamics (6) possesses the desired characteristics. For instance, these matrices are typically chosen to be constant and diagonal, in which case the dynamical system (6) represents n uncoupled second-order systems in the n task-space coordinates. It can be seen from (6) that, in the absence of environmental contact, we have $\mathbf{P} = \mathbf{0}$ and $\mathbf{x}(t) = \mathbf{x}_r(t) \;\; \forall t$ (assuming $\mathbf{x} = \mathbf{x}_r$ initially), which corresponds to free-space trajectory tracking. In the presence of contact, the desired response is for the end-effector to comply with the contact forces in the manner defined by the impedance model (6).

The proposed impedance control system consists of two subsystems: a simple filter which characterizes the desired performance (6), and an adaptive task-space control scheme which generates the control input $\mathbf{u}$ to the RLED manipulator system (5) that ensures this desired performance is realized. In what follows, the proposed control system is presented and analyzed, and then simulation results illustrating its performance are described.

3.1 Control Scheme

Observe that the impedance control objective can be realized if the end-effector position $\mathbf{x}$ closely tracks the *desired impedance trajectory* $\mathbf{x}_d \in \Re^n$, defined as the solution to the differential equation:

$$\begin{aligned} M_{imp}\ddot{\mathbf{x}}_d + C_{imp}\dot{\mathbf{x}}_d + K_{imp}\mathbf{x}_d &= -\mathbf{P} + M_{imp}\ddot{\mathbf{x}}_r + C_{imp}\dot{\mathbf{x}}_r + K_{imp}\mathbf{x}_r \\ \mathbf{x}_d(0) = \mathbf{x}_r(0) \quad &, \quad \dot{\mathbf{x}}_d(0) = \dot{\mathbf{x}}_r(0) \end{aligned} \tag{7}$$

obtained from the impedance model (6). Equation (7) may be interpreted as defining a simple filter which, given definitions for $\mathbf{x}_r(t), M_{imp}, C_{imp}$, and K_{imp} and on-line measurements $\mathbf{P}$, characterizes the desired dynamic relationship between end-effector position and contact force through the specification of $\mathbf{x}_d$. The function of the task-space impedance controller is to cause the RLED manipulator system (5) to evolve in such a way that the end-effector position $\mathbf{x}$ closely tracks $\mathbf{x}_d$ defined in (7).

Let $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ denote the trajectory tracking error and $\mathbf{E} = \mathbf{T}_d - \mathbf{T}$ represent the torque tracking error. Consider the following adaptive tracking scheme for RLED manipulators:

$$\begin{aligned} \mathbf{F}_d &= A(t)\dot{\mathbf{x}}_d + \mathbf{f}(t) + \mathbf{P} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \\ \dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \\ \mathbf{T}_d &= J_a^T\mathbf{F}_d \\ \mathbf{u} &= [\mathbf{a}(t)\dot{\mathbf{T}}_d] + [\mathbf{b}(t)\mathbf{T}] + [\mathbf{c}(t)\dot{\theta}] + k_a\mathbf{E} + J_a^{-1}\mathbf{q} \end{aligned} \tag{8}$$

where the notation $[\mathbf{gh}] = [g_1h_1, g_2h_2, ..., g_nh_n]^T \in \Re^n$ (for any two n-vectors $\mathbf{g}, \mathbf{h}$) is introduced, $\mathbf{q} = \dot{\mathbf{e}} + k_2\mathbf{e}/k_1\gamma - \mathbf{w}/\gamma$ is the weighted and filtered position-velocity error, k_1, k_2, γ, k_a are positive scalar constants, and $\mathbf{f}(t), \mathbf{a}(t), \mathbf{b}(t), \mathbf{c}(t) \in \Re^n$, $A(t) \in \Re^{n\times n}$ are adaptive gains which are adjusted according to the following simple update laws:

$$\dot{\mathbf{f}} = -\sigma_1\mathbf{f} + \beta_1\mathbf{q}$$

$$
\begin{aligned}
\dot{A} &= -\sigma_2 A + \beta_2 \mathbf{q}\dot{\mathbf{x}}_d^T \\
\dot{\mathbf{a}} &= -\sigma_3 \mathbf{a} + \beta_3 [\mathbf{E}\dot{\mathbf{T}}_d] \\
\dot{\mathbf{b}} &= -\sigma_4 \mathbf{b} + \beta_4 [\mathbf{E}\mathbf{T}] \\
\dot{\mathbf{c}} &= -\sigma_5 \mathbf{c} + \beta_5 [\mathbf{E}\dot{\theta}]
\end{aligned}
\tag{9}
$$

where the σ_i and β_i are positive scalar adaptation gains. Inspection of (7)-(9) reveals that both $\mathbf{T}_d$ and $\dot{\mathbf{T}}_d$ can be easily and efficiently computed using only measurements of manipulator joint position and velocity and end-effector contact force. As a consequence, the complete controller is computationally efficient and readily implementable using measurements of θ, $\dot{\theta}$, $\mathbf{T}$, and $\mathbf{P}$ (recall that the same design procedure can be used for the case in which θ, $\dot{\theta}$, $\mathbf{I}$, and $\mathbf{P}$ are measurable).

The stability properties of the proposed adaptive impedance controller (7)-(9) are summarized in the following theorem.

Theorem 1: The adaptive scheme (7)-(9) ensures that (5) evolves with all signals (semiglobally) uniformly bounded provided γ is chosen sufficiently large. Moreover, the impedance control error $\mathbf{e}, \dot{\mathbf{e}}$ is guaranteed to converge exponentially to a compact set which can be made arbitrarily small.

Proof: Applying the control law (8) to the manipulator dynamics (5) yields the closed-loop error dynamics

$$
\begin{gathered}
H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi}_f + \Phi_A\dot{\mathbf{x}}_d + V_{cd}\dot{\mathbf{e}} - J_a^{-T}\mathbf{E} = \mathbf{0} \\
[\mathbf{m}\dot{\mathbf{E}}] + k_a\mathbf{E} + J_a^{-1}\mathbf{q} + [\phi_a\dot{\mathbf{T}}_d] + [\phi_b\mathbf{T}] + [\phi_c\dot{\theta}] = \mathbf{0}
\end{gathered}
\tag{10}
$$

where $\mathbf{m}, \mathbf{r}, \mathbf{k} \in \Re^n$ are formed by stacking the n diagonal elements of M, R, K into vectors, $\phi_a = \mathbf{a} - \mathbf{m}$, $\phi_b = \mathbf{b} - \mathbf{r}$, $\phi_c = \mathbf{c} - \mathbf{k}$, $\mathbf{\Phi}_f = \mathbf{f} - H\ddot{\mathbf{x}}_d - \mathbf{G}$, $\Phi_A = A - V_{cd}$, and the notation $V_{cd} = V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d)$ is introduced. Note that in obtaining (10) the Coriolis/centripetal term was expanded as follows: $V_{cc}\dot{\mathbf{x}}_d = V_{cd}\dot{\mathbf{x}} = V_{cd}\dot{\mathbf{x}}_d - V_{cd}\dot{\mathbf{e}}$.

Consider the Lyapunov function candidate

$$
\begin{aligned}
V_1 = {} & \frac{1}{2}\dot{\mathbf{e}}^T H\dot{\mathbf{e}} + \frac{1}{2}k_2\gamma^2\mathbf{e}^T\mathbf{e} + \frac{1}{2}k_1\mathbf{w}^T\mathbf{w} + \frac{k_2}{k_1\gamma}\mathbf{e}^T H\dot{\mathbf{e}} - \frac{1}{\gamma}\mathbf{w}^T H\dot{\mathbf{e}} + \frac{1}{2}\mathbf{E}^T[\mathbf{m}\mathbf{E}] \\
& + \frac{1}{2\beta_1}\mathbf{\Phi}_f^T\mathbf{\Phi}_f + \frac{1}{2\beta_2}\mathrm{tr}[\Phi_A\Phi_A^T] + \frac{1}{2\beta_3}\phi_a^T\phi_a + \frac{1}{2\beta_4}\phi_b^T\phi_b + \frac{1}{2\beta_5}\phi_c^T\phi_c
\end{aligned}
\tag{11}
$$

and note that V_1 is a positive-definite and radially-unbounded function of the closed-loop system state if γ is chosen sufficiently large. Computing the derivative of (11) along (10), and simplifying using (9) and the definition for $\mathbf{w}$, yields

$$
\dot{V}_1 = -(\gamma - \frac{k_2}{k_1\gamma})\dot{\mathbf{e}}^T H\dot{\mathbf{e}} - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 2 - k_1\gamma \parallel \mathbf{w} \parallel^2 - k_a \parallel \mathbf{E} \parallel^2
$$

$$
\begin{aligned}
& + 2\mathbf{w}^T H\dot{\mathbf{e}} + \frac{1}{\gamma}\dot{\mathbf{e}}^T[\frac{k_2}{k_1}V_{cc}\mathbf{e} - V_{cc}\mathbf{w}] - \mathbf{q}^T V_{cd}\dot{\mathbf{e}} \\
& + \frac{1}{\beta_1}\mathbf{\Phi}_f^T(\dot{\mathbf{\Phi}}_f - \beta_1\mathbf{q}) + \frac{1}{\beta_2}\mathrm{tr}[\Phi_A(\dot{\Phi}_A - \beta_2\mathbf{q}\dot{\mathbf{x}}_d^T)^T] \\
& + \frac{1}{\beta_3}\phi_a^T(\dot{\phi}_a - \beta_3[\mathbf{E}\dot{\mathbf{T}}_d]) + \frac{1}{\beta_4}\phi_b^T(\dot{\phi}_b - \beta_4[\mathbf{E}\mathbf{T}]) + \frac{1}{\beta_5}\phi_c^T(\dot{\phi}_c - \beta_5[\mathbf{E}\dot{\theta}]) \\
\leq & -((\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_m(H) - k_{cd}) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 - k_a \parallel \mathbf{E} \parallel^2 \\
& + (2\lambda_M(H) + \frac{k_{cd}}{\gamma} + \frac{k_{cc}v_M}{\gamma}) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{k_2}{k_1\gamma}(k_{cc}v_M + k_{cd}) \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel \\
& + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma_1}{2\beta_1} \parallel \mathbf{\Phi}_f \parallel^2 - \frac{\sigma_2}{2\beta_2} \parallel \Phi_A \parallel_F^2 \\
& - \frac{\gamma}{4}\lambda_m(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma_1}{2\beta_1} \parallel \mathbf{\Phi}_f \parallel^2 + \frac{\eta_1}{\beta_1} \parallel \dot{\mathbf{e}} \parallel\parallel \mathbf{\Phi}_f \parallel + \frac{\eta_2}{\beta_1} \parallel \mathbf{\Phi}_f \parallel \\
& - \frac{\gamma}{4}\lambda_m(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma_2}{2\beta_2} \parallel \Phi_A \parallel_F^2 + \frac{\eta_3}{\beta_2} \parallel \dot{\mathbf{e}} \parallel\parallel \Phi_A \parallel_F + \frac{\eta_4}{\beta_2} \parallel \Phi_A \parallel_F \\
& - \frac{\sigma_3}{\beta_3} \parallel \phi_a \parallel^2 - \frac{\sigma_4}{\beta_4} \parallel \phi_b \parallel^2 - \frac{\sigma_5}{\beta_5} \parallel \phi_c \parallel^2 + \frac{\eta_5}{\beta_3} + \frac{\eta_6}{\beta_4} + \frac{\eta_7}{\beta_5} \qquad (12)
\end{aligned}
$$

where $\lambda_m(\cdot), \lambda_M(\cdot)$ denote the minimum and maximum eigenvalue of the matrix argument, respectively, $\parallel \cdot \parallel_F$ is the Frobenius matrix norm, k_{cc} satisfies $\parallel V_{cc} \parallel_F \leq k_{cc} \parallel \dot{\mathbf{x}} \parallel \forall \mathbf{x}$, k_{cd} is an upper bound on V_{cd} (i.e., $\parallel V_{cd} \parallel_F \leq k_{cd} \;\; \forall \mathbf{x}$), v_M is an upper bound on $\parallel \dot{\mathbf{x}}_d \parallel$, and the η_i are positive scalar constants. Note that the existence of v_{max} can be concluded by examining (7) and noticing that $\mathbf{x}_r$, $\dot{\mathbf{x}}_r$, $\ddot{\mathbf{x}}_r$ are bounded by definition and $\mathbf{P}$ is bounded because the workspace of the manipulator is assumed to be finite (so that the environmental deformation must be finite) [22,23].

Observe that the following inequalities can be obtained through routine manipulation:

$$
-\frac{\gamma}{4}\lambda_m(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma_1}{2\beta_1} \parallel \mathbf{\Phi}_f \parallel^2 + \frac{\eta_1}{\beta_1} \parallel \dot{\mathbf{e}} \parallel\parallel \mathbf{\Phi}_f \parallel + \frac{\eta_2}{\beta_1} \parallel \mathbf{\Phi}_f \parallel \leq \frac{\eta_8}{\beta_1}
$$

$$
-\frac{\gamma}{4}\lambda_m(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma_2}{2\beta_2} \parallel \Phi_A \parallel_F^2 + \frac{\eta_3}{\beta_2} \parallel \dot{\mathbf{e}} \parallel\parallel \Phi_A \parallel_F + \frac{\eta_4}{\beta_2} \parallel \Phi_A \parallel_F \leq \frac{\eta_9}{\beta_2}
$$

where η_8, η_9 are positive scalar constants which do not increase as the β_i are increased. These inequalities permit the following upper bound on $\dot{V}_1$ in (12) to be established:

$$
\begin{aligned}
\dot{V}_1 \leq & -((\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_m(H) - k_{cd}) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 - k_a \parallel \mathbf{E} \parallel^2 \\
& + (2\lambda_M(H) + \frac{k_{cd}}{\gamma} + \frac{k_{cc}v_M}{\gamma}) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{k_2}{k_1\gamma}(k_{cc}v_M + k_{cd}) \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel \\
& + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma_1}{2\beta_1} \parallel \mathbf{\Phi}_f \parallel^2 - \frac{\sigma_2}{2\beta_2} \parallel \Phi_A \parallel_F^2 \\
& - \frac{\sigma_3}{\beta_3} \parallel \phi_a \parallel^2 - \frac{\sigma_4}{\beta_4} \parallel \phi_b \parallel^2 - \frac{\sigma_5}{\beta_5} \parallel \phi_c \parallel^2 + \frac{\eta_8}{\beta_1} + \frac{\eta_9}{\beta_2} + \frac{\eta_5}{\beta_3} + \frac{\eta_6}{\beta_4} + \frac{\eta_7}{\beta_5} \qquad (13)
\end{aligned}
$$

Let $\mathbf{z}_1 = [\| \mathbf{e} \| \quad \| \dot{\mathbf{e}} \| \quad \| \mathbf{w} \|]^T$, $\boldsymbol{\Psi} = [\| \boldsymbol{\Phi}_f \| \quad \| \boldsymbol{\Phi}_A \|_F \quad \| \phi_a \| \quad \| \phi_b \| \quad \| \phi_c \|]^T$, $\beta_{min} = \min(\beta_i)$, $\beta_{max} = \max(\beta_i)$, and

$$Q^* = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} & -\frac{k_2}{2k_1\gamma}(k_{cc}v_M + k_{cd}) & 0 \\ -\frac{k_2}{2k_1\gamma}(k_{cc}v_M + k_{cd}) & (\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_m(H) - k_{cd} & -\lambda_M(H) - \frac{k_{cd}}{2\gamma} - \frac{k_{cc}v_M}{2\gamma} \\ 0 & -\lambda_M(H) - \frac{k_{cd}}{2\gamma} - \frac{k_{cc}v_M}{2\gamma} & k_1\gamma \end{bmatrix}$$

and note that Q^* is positive-definite if γ is chosen large enough. With these definitions the following bound on $\dot{V}_1$ in (13) can be derived:

$$\begin{aligned} \dot{V}_1 \leq & - \lambda_m(Q^*) \| \mathbf{z}_1 \|^2 - k_a \| \mathbf{E} \|^2 - \frac{\eta_{10}}{\beta_{min}} \| \boldsymbol{\Psi} \|^2 \\ & + \frac{k_2 k_{cc}}{k_1\gamma} \| \mathbf{e} \| \| \dot{\mathbf{e}} \|^2 + \frac{k_{cc}}{\gamma} \| \mathbf{w} \| \| \dot{\mathbf{e}} \|^2 + \frac{\eta_{11}}{\beta_{min}} \end{aligned} \tag{14}$$

where η_{10}, η_{11} are scalar constants. Finally, let $\mathbf{z}_2 = [\| \mathbf{z}_1 \| \quad \| \mathbf{E} \|]^T$ and $Q = \text{diag}[\lambda_m(Q^*), k_a]$. Then if β_{max}/β_{min} is fixed there exists a positive scalar constant η_{12} that does not increase as γ and β_{min} increase, and positive scalar constants λ_i independent of γ and β_{min}, such that V_1 and $\dot{V}_1$ in (11) and (14) can be bounded as

$$\begin{aligned} \lambda_1 \| \mathbf{z}_2 \|^2 + \frac{\lambda_2}{\beta_{min}} \| \boldsymbol{\Psi} \|^2 \leq V_1 \leq \lambda_3 \| \mathbf{z}_2 \|^2 + \frac{\lambda_4}{\beta_{min}} \| \boldsymbol{\Psi} \|^2 \\ \dot{V}_1 \leq -(\lambda_m(Q) - \frac{\eta_{12}}{\gamma} {V_1}^{1/2}) \| \mathbf{z}_2 \|^2 - \frac{\lambda_5}{\beta_{min}} \| \boldsymbol{\Psi} \|^2 + \frac{\eta_{11}}{\beta_{min}} \end{aligned}$$

Now choose two scalar constants V_M, V_m so that $V_M > V_m \geq V_1(0)$, and define $c_M = \lambda_m(Q) - \eta_{12}{V_M}^{1/2}/\gamma$; then choose γ large enough so that $c_M > 0$ (this is always possible). Let $\delta = \max(\lambda_3/c_M, \lambda_4/\lambda_5)$ and choose β_0 so that $\eta_{11}\delta/\beta_0 < V_m$ (this is always possible). Then selecting $\beta_{min} \geq \beta_0$ ensures that if $V_m \leq V_1 \leq V_M$ then $\dot{V}_1 < 0$. This condition together with $V_M > V_m \geq V_1(0)$ implies that $V_1(t) \leq V_M \; \forall t$, so that $c(t) = \lambda_m(Q) - \eta_{12}V_1^{1/2}(t)/\gamma > c_M > 0 \;\; \forall t$ and

$$\dot{V}_1 \leq -c_M \| \mathbf{z}_2 \|^2 - \frac{\lambda_5}{(\beta_0 + \Delta\beta)} \| \boldsymbol{\Psi} \|^2 + \frac{\eta_{11}}{(\beta_0 + \Delta\beta)}$$

where $\Delta\beta = \beta_{min} - \beta_0$ and it is assumed that β_{min} is chosen so that $\Delta\beta > 0$. Lemma UB now applies and permits the conclusion that $\| \mathbf{z}_2 \|, \| \boldsymbol{\Psi} \|$ are uniformly bounded, which implies that $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$, $\mathbf{E}$, $\mathbf{f}$, A, $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are uniformly bounded. Moreover, $\| \mathbf{z}_2 \|, \| \boldsymbol{\Psi} \|$ converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where

$$r_1 = \left(\frac{\delta\eta_{11}}{\lambda_1(\beta_0 + \Delta\beta)} \right)^{1/2}$$

$$r_2 = \left(\frac{\delta\eta_{11}}{\lambda_2} \right)^{1/2}$$

Observe that the radius of the ball to which $\| \mathbf{z}_2 \|^2$ is guaranteed to converge can be decreased as desired simply by increasing $\Delta\beta$. ■

Several observations can be made concerning the adaptive impedance controller (7)-(9). First note that the control strategy is simple, utilizes only measurements of the system state and contact force, and requires virtually no *a priori* information concerning the manipulator, the actuators, or the environment. Thus the proposed scheme provides a computationally efficient, modular, and readily implementable approach to RLED manipulator compliant motion control. Theorem 1 shows that the controller ensures uniform boundedness of all signals, and that the ultimate bounds on the discrepancy between the desired and actual performance can be reduced arbitrarily by increasing the adaptation gains β_i. Note that increasing these adaptation gains does not ordinarily lead to large control action. To see this, observe from the ultimate bound on $\| \Psi \|$ that increasing the β_i does not increase the size of the difference between adaptive terms and model terms, and therefore does not cause unwarranted growth in the control law gains. It is mentioned that the convergence of the system errors is exponential, which ensures that the transient behavior of the system will be well-behaved. Finally, note that the stability properties established in Theorem 1 are *semiglobal* in the sense that the region of attraction can be increased arbitrarily by increasing the controller gain γ. Again it is stressed that this does not imply that γ must be chosen to be overly large; indeed, we have found that excellent performance is obtainable with this gain set to quite modest values.

3.2 Simulation Results

The adaptive impedance control scheme described above is now applied to a six DOF robot manipulator through computer simulation. The robot chosen for this simulation study is the IMI Zebra Zero robot; this manipulator possesses a conventional design with six revolute joints configured in a "waist-shoulder-elbow-wrist" arrangement (see Figure 1). The simulation environment incorporates models of all important dynamic subsystems and phenomena, such as the full nonlinear arm dynamics, actuator dynamics, joint stiction, sensor noise, and transmission effects, and therefore provides the basis for a realistic evaluation of controller performance. The control law is applied to the manipulator model with a sampling period of two milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of two milliseconds.

This simulation study demonstrates that the proposed impedance control scheme provides a unified framework for unconstrained and constrained motion control, and illustrates how the scheme can accommodate the transition between unconstrained and constrained end-effector motion. In this study, a frictionless reaction surface with a stiffness of 10^5 N/m is placed in the robot workspace. This reaction surface is oriented normal to the x axis of the robot base frame and is located at $x = 1.25$m. The reference trajectory for end-effector translation is defined as $x_r(t) = 1.0+0.5\sin\pi t/4, y_r(t) = -1.0+0.5(1-\cos\pi t/4), z_r(t) = 0.0$ for $t \in [0, 4]$. Additionally, the manipulator is required to maintain its initial end-effector orientation. The end-effector reference trajectory therefore consists of a rapid semicircular motion in the x, y plane with a radius of 0.5m, as shown in Figure 2. Note that the reference trajectory intersects the reaction surface at approximately $t = 0.7$s, so that at this point the robot undergoes an abrupt transition from unconstrained to constrained motion.

The adaptive impedance controller (7)-(9) is used in this study. The impedance parameters in (7) are defined as $M_{imp} = \text{diag}[0.5], K_{imp} = \text{diag}[100]$, and $C_{imp} = \text{diag}[2\sqrt{m_{ii}k_{ii}}]$; observe that the elements of C_{imp} are set to provide critical damping in the impedance model. The adaptive gains $\mathbf{f}$, A, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ are set to zero initially, while the remaining controller parameters are set as follows: $k_1 = 10$, $k_2 = 20$, $\gamma = 5$, $k_a = 100$ and $\sigma_i = 0.01$, $\beta_i = 100$ for $i = 1, 2, ..., 5$. The results of this simulation are given in Figures 2a and 2b. Figure 2a indicates that the end-effector closely tracks the reference trajectory until contact with the reaction surface is made. At contact, the robot smoothly ceases to track the x component of the reference trajectory in favor of complying with the reaction surface; note that this behavior has very little effect on the robot motion in the y direction. The end-effector/surface contact force $\mathbf{P}$ is shown in Figure 2b. It can be seen that the transient forces incurred during the transition from unconstrained to constrained motion are well behaved, and that the contact force follows a semicircular profile, reflecting the influence of the user-specified reference trajectory on the contact force.

4. Adaptive Position/Force Controller

We now turn to the development of an adaptive position/force controller for uncertain RLED manipulators. Consider the manipulator/actuator system (5) interacting with a linearly elastic environment. In what follows we shall model the rigid manipulator portion of this system using the approach proposed in [23]. Thus we suppose the existence of a *task frame* T, and let $\mathbf{x}$ denote the (possibly augmented) end-effector coordinates relative to this frame. It is easy to show that all of the "nice" properties of the rigid-link manipulator dynamics (5a) which are summarized in Section 2 are retained with this definition for the system task-space coordinate vector $\mathbf{x}$ [24]. Moreover, in this case the end-effector/environment interaction can be modeled as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_t \\ \mathbf{P}_t' \end{bmatrix} = \mathcal{K}_e(\mathbf{x}\ 2 - \mathbf{x}5_e3) = \begin{bmatrix} K_e \\ K_e' \end{bmatrix} (\mathbf{x} - \mathbf{x}_e) \qquad (15)$$

In (15), $\mathcal{K}_e \in \Re^{n\times n}$ and $\mathbf{x}_e \in \Re^n$ are the environmental stiffness and location, respectively, $\mathbf{P}_t \in \Re^r$, $K_e \in \Re^{r\times n}$ has independent rows, and the elements of $\mathbf{P}'_t$ and rows of K'_e are linear combinations of the elements of $\mathbf{P}_t$ and rows of K_e, respectively; therefore only the $\mathbf{P}_t$ component of $\mathbf{P}$ can be independently controlled. Let $K_e^+ = K_e^T(K_eK_e^T)^{-1}$, $K_e^- = I - K_e^+K_e$, and define $K_e^* \in \Re^{n\times(n-r)}$ and $\mathbf{x}_t \in \Re^{n-r}$ so that the range spaces of K_e^- and K_e^* are identical and so that $K_e^*\dot{\mathbf{x}}_t = K_e^-\dot{\mathbf{x}}$ $\forall\dot{\mathbf{x}}$. With these definitions it is easily verified that $\dot{\mathbf{x}} = K_e^+\dot{\mathbf{P}}_t + K_e^*\dot{\mathbf{x}}_t$, so that $\mathbf{P}_t$, $\mathbf{x}_t$ completely define the configuration of the rigid-link manipulator (5a) and can be independently controlled [23]. The complete RLED manipulator model (5) can be written in terms of the pair $\mathbf{P}_t$, $\mathbf{x}_t$ as follows:

$$\mathbf{F} = H(K_e^+\ddot{\mathbf{P}}_t + K_e^*\ddot{\mathbf{x}}_t) + V_{cc}(K_e^+\dot{\mathbf{P}}_t + K_e^*\dot{\mathbf{x}}_t) + \mathbf{G} + \mathbf{P}, \qquad \mathbf{T} = J_a^T\mathbf{F} \tag{16a}$$

$$\mathbf{u} = M\dot{\mathbf{T}} + R\mathbf{T} + K\dot{\theta} \tag{16b}$$

In typical applications the environmental stiffness matrix $\mathcal{K}_e$ is not accurately known. However, if the environment is isotropic in addition to elastic, $\mathcal{K}_e$ can be parameterized as $\mathcal{K}_e = k_eE$, where k_e is an unknown positive constant and $E \in \Re^{n\times n}$ is easily determined from the environmental geometry. In this case, direct calculation reveals that $K_e^+ = (1/k_e)E^+$ and $K_e^* = E^*$, so that K_e^* is known and the only uncertainty associated with K_e^+ is an unknown constant coefficient; it will be shown below that this latter uncertainty can be compensated with the proposed control scheme.

4.1 Control Scheme

The objective of the proposed position/force controller is to ensure that the RLED manipulator dynamics (16) evolves so that $\mathbf{x}_t$ tracks the desired position trajectory $\mathbf{x}_{td}(t)$ (which is bounded with bounded derivatives) and $\mathbf{P}_t$ approaches the desired constant force setpoint $\mathbf{P}_{td}$. Consider the following adaptive solution to this problem:

$$\begin{aligned}
\mathbf{F}_d &= A(t)\ddot{\mathbf{x}}_{td} + B(t)\dot{\mathbf{x}}_{td} + \mathbf{f}(t) + \mathbf{P} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e}\\
\dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}}\\
\dot{\mathbf{f}} &= -\sigma_1\mathbf{f} + \beta_1\mathbf{q}\\
\dot{A} &= -\sigma_2A + \beta_2\mathbf{q}\ddot{\mathbf{x}}_{td}^T\\
\dot{B} &= -\sigma_3B + \beta_3\mathbf{q}\dot{\mathbf{x}}_{td}^T\\
\mathbf{T}_d &= J_a^T\mathbf{F}_d\\
\mathbf{u} &= [\mathbf{a}(t)\dot{\mathbf{T}}_d] + [\mathbf{b}(t)\mathbf{T}] + [\mathbf{c}(t)\dot{\theta}] + k_a\mathbf{E} + J_a^{-1}\mathbf{q}\\
\dot{\mathbf{a}} &= -\sigma_4\mathbf{a} + \beta_4[\mathbf{E}\dot{\mathbf{T}}_d]\\
\dot{\mathbf{b}} &= -\sigma_5\mathbf{b} + \beta_5[\mathbf{E}\mathbf{T}]\\
\dot{\mathbf{c}} &= -\sigma_6\mathbf{c} + \beta_6[\mathbf{E}\dot{\theta}]
\end{aligned} \tag{17}$$

where $\mathbf{e} = (1/k_e)E^+(\mathbf{P}_{td} - \mathbf{P}_t) + E^*(\mathbf{x}_{td} - \mathbf{x}_t)$ is the position/force error. Inspection of (17) reveals that both $\mathbf{T}_d$ and $\dot{\mathbf{T}}_d$ can be easily and efficiently computed using only measurements of manipulator joint position and velocity and end-effector contact force. As a consequence, the complete controller is computationally efficient and readily implementable using measurements of θ, $\dot\theta$, $\mathbf{T}$, and $\mathbf{P}$ (recall that the same design procedure can be used for the case in which θ, $\dot\theta$, $\mathbf{I}$, and $\mathbf{P}$ are measurable).

The stability properties of the proposed adaptive position/force controller (17) are summarized in the following theorem.

Theorem 2: The adaptive controller (17) ensures that (16) evolves with all signals (semiglobally) uniformly bounded provided γ is chosen sufficiently large. Moreover, the position error $\mathbf{x}_{td} - \mathbf{x}_t$ and force error $\mathbf{P}_{td} - \mathbf{P}_t$ are guaranteed to converge exponentially to compact sets which can be made arbitrarily small.

Proof: Applying the control law (17) to the RLED manipulator dynamics (16) yields the closed-loop system dynamics

$$\begin{aligned} &H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi}_f^* + \Phi_A^*\ddot{\mathbf{x}}_{td} + \Phi_B\dot{\mathbf{x}}_{td} + V_{cd}\dot{\mathbf{e}} - J_a^{-T}\mathbf{E} = \mathbf{0} \\ &[\mathbf{m}\dot{\mathbf{E}}] + k_a\mathbf{E} + J_a^{-1}\mathbf{q} + [\phi_a\dot{\mathbf{T}}_d] + [\phi_b\mathbf{T}] + [\phi_c\dot\theta] = \mathbf{0} \end{aligned} \tag{18}$$

where $\mathbf{\Phi}_f^* = \mathbf{f} - \mathbf{G}$, $\Phi_A^* = A - HE^*$, $\Phi_B = B - V_{cd}E^*$, and all other terms are defined as before. Consider the Lyapunov function candidate

$$\begin{aligned} V_2 = {} & \frac{1}{2}\dot{\mathbf{e}}^TH\dot{\mathbf{e}} + \frac{1}{2}k_2\gamma^2\mathbf{e}^T\mathbf{e} + \frac{1}{2}k_1\mathbf{w}^T\mathbf{w} + \frac{k_2}{k_1\gamma}\mathbf{e}^TH\dot{\mathbf{e}} - \frac{1}{\gamma}\mathbf{w}^TH\dot{\mathbf{e}} + \frac{1}{2}\mathbf{E}^T[\mathbf{mE}] \\ & + \frac{1}{2\beta_1}\mathbf{\Phi}_f^{*T}\mathbf{\Phi}_f^* + \frac{1}{2}\mathrm{tr}[\frac{1}{\beta_2}\Phi_A^*\Phi_A^{*T} + \frac{1}{\beta_3}\Phi_B\Phi_B^T] + \frac{1}{2\beta_4}\phi_a^T\phi_a + \frac{1}{2\beta_5}\phi_b^T\phi_b + \frac{1}{2\beta_6}\phi_c^T\phi_c \end{aligned} \tag{19}$$

and note that V_2 is a positive-definite and radially-unbounded function of the closed-loop system state if γ is chosen sufficiently large. Computing the derivative of (19) along (18) and simplifying as in the proof of Theorem 1 yields the following upper bound on $\dot V_2$:

$$\begin{aligned} \dot V_2 \le & -\lambda_m(Q^*) \parallel \mathbf{z}_1 \parallel^2 - k_a \parallel \mathbf{E} \parallel^2 - \frac{\eta_{13}}{\beta_{min}} \parallel \mathbf{\Psi}^* \parallel^2 \\ & + \frac{k_2k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{\eta_{14}}{\beta_{min}} \end{aligned} \tag{20}$$

where $\mathbf{\Psi}^* = [\parallel \mathbf{\Phi}_f^* \parallel \ \ \parallel \Phi_A^* \parallel_F \ \ \parallel \Phi_B \parallel_F \ \ \parallel \phi_a \parallel \ \ \parallel \phi_b \parallel \ \ \parallel \phi_c \parallel]^T$ and η_{13}, η_{14} are scalar constants.

Now arguments identical to the ones used in the proof of Theorem 1 can be used to show that $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$, $\mathbf{E}$, $\mathbf{f}$, A, B, $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are uniformly bounded, and that the ultimate size of the ball to which $\mathbf{e}$ and $\dot{\mathbf{e}}$ are guaranteed to converge can be decreased as desired simply by increasing the β_i. Observe that this latter fact implies that the ultimate size of the position error $\mathbf{x}_{td} - \mathbf{x}_t$ and force error $\mathbf{P}_{td} - \mathbf{P}_t$ can be reduced arbitrarily as well. To see this, note that small $\mathbf{e}$ implies that $K_e \mathbf{e} = \mathbf{P}_{td} - \mathbf{P}_t$ is small, and this in turn implies that $\mathbf{x}_{td} - \mathbf{x}_t$ is small. ■

Several observations can be made concerning the proposed adaptive position/force controller (17). First note that the control strategy is simple, utilizes only system state and force measurements, and requires virtually no information concerning either the manipulator/actuator system dynamics or the environment. In particular, it is noted that knowledge of the environmental stiffness is not required for implementing the scheme, despite the appearance of k_e in the definition of position/force error $\mathbf{e}$. To see this, notice that $\mathbf{e}$ always appears with constant coefficients in the control strategy, so that k_e can always be "absorbed" into these constants for implementation. Thus the scheme provides a computationally efficient, modular, and readily implementable approach to RLED manipulator compliant motion control. Theorem 2 shows that the controller ensures uniform boundedness of all signals, and that the ultimate bounds on the discrepancy between the desired and actual performance can be reduced arbitrarily by increasing the adaptation gains β_i. Thus all of the observations made regarding the impedance controller (7)-(9) following the proof of Theorem 1 apply to the position/force control scheme (17) as well.

Finally, it is interesting to observe that, if $\mathbf{x}_{td}$ is a constant position setpoint, then the proposed adaptive controller (17) yields *asymptotic* position/force regulation. This result is established in the following corollary to Theorem 2.

Corollary: Suppose that the desired position trajectory $\mathbf{x}_{td}$ is a constant setpoint. Then, if the σ_i are set to zero, the adaptive controller (17) ensures that $\mathbf{x}_t \to \mathbf{x}_{td}$ and $\mathbf{P}_t \to \mathbf{P}_{td}$ as $t \to \infty$.

Proof: Applying the control law (17) (with $\dot{\mathbf{x}}_{td} = \ddot{\mathbf{x}}_{td} = \mathbf{0}$) to the manipulator dynamics (16) yields the closed-loop system dynamics

$$\begin{aligned} &H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi}_d + \mathbf{G}(\mathbf{x}_{td}) - \mathbf{G}(\mathbf{x}_t) - J_a^{-T}\mathbf{E} = \mathbf{0} \\ &[\mathbf{m}\dot{\mathbf{E}}] + k_a\mathbf{E} + J_a^{-1}\mathbf{q} + [\phi_a\dot{\mathbf{T}}_d] + [\phi_b\mathbf{T}] + [\phi_c\dot{\theta}] = \mathbf{0} \end{aligned} \tag{21}$$

where $\mathbf{\Phi}_d = \mathbf{f} - \mathbf{G}(\mathbf{x}_{td})$. Let $U(\mathbf{x}_t)$ denote the manipulator gravitational potential energy. Consider the Lyapunov function candidate

$$\begin{aligned} V_3 = &\frac{1}{2}\dot{\mathbf{e}}^T H\dot{\mathbf{e}} + \frac{1}{2}k_2\gamma^2\mathbf{e}^T\mathbf{e} + \frac{1}{2}k_1\mathbf{w}^T\mathbf{w} + \frac{k_2}{k_1\gamma}\mathbf{e}^T H\dot{\mathbf{e}} - \frac{1}{\gamma}\mathbf{w}^T H\dot{\mathbf{e}} + \frac{1}{2}\mathbf{E}^T[\mathbf{m}\mathbf{E}] \\ &+ \frac{1}{2\beta_1}\mathbf{\Phi}_d^T\mathbf{\Phi}_d + \frac{1}{2\beta_4}\phi_a^T\phi_a + \frac{1}{2\beta_5}\phi_b^T\phi_{b}2+56\frac{1}{2\beta_6}\phi_c^T\phi_c + U(\mathbf{x}_t) - U(\mathbf{x}_{td}) + \mathbf{G}^T(\mathbf{x}_{td})\mathbf{e} \end{aligned} \tag{22}$$

and note that V_3 is a positive-definite and radially-unbounded function of the closed-loop system state if γ is chosen sufficiently large (for example, if γ is chosen large enough then the term $k_2\gamma^2\mathbf{e}^T\mathbf{e}/2 + U(\mathbf{x}_t) - U(\mathbf{x}_{td}) + \mathbf{G}^T(\mathbf{x}_{td})\mathbf{e}$ can be shown to be positive-definite in $\mathbf{e}$).

Differentiating (22) along (21) and simplifying as in the proof of Theorem 2 yields

$$
\begin{aligned}
\dot{V}_3 &= -(\gamma - \frac{k_2}{k_1\gamma})\dot{\mathbf{e}}^T H\dot{\mathbf{e}} - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 - k_a \parallel \mathbf{E} \parallel^2 + 2\mathbf{w}^T H\dot{\mathbf{e}} \\
&\quad + \frac{1}{\gamma}\dot{\mathbf{e}}^T[\frac{k_2}{k_1}V_{cc}\mathbf{e} - V_{cc}\mathbf{w}] + \frac{1}{\beta_1}\mathbf{\Phi}_d^T(\dot{\mathbf{f}} - \beta_1\mathbf{q}) + \frac{1}{\beta_4}\phi_a^T(\dot{\mathbf{a}} - \beta_4[\mathbf{E}\dot{\mathbf{T}}_d]) \\
&\quad + \frac{1}{\beta_5}\phi_b^T(\dot{\mathbf{b}} - \beta_5[\mathbf{ET}]) + \frac{1}{\beta_6}\phi_c^T(\dot{\mathbf{c}} - \beta_6[\mathbf{E}\dot{\theta}]) + (\mathbf{G}(\mathbf{x}_t) - \mathbf{G}(\mathbf{x}_{td}))(\frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}) \\
&\leq -(\gamma - \frac{k_2}{k_1\gamma})\lambda_m(H) \parallel \dot{\mathbf{e}} \parallel^2 - (\frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma}) \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 - k_a \parallel \mathbf{E} \parallel^2 \qquad (23) \\
&\quad + 2\lambda_M(H) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{M}{\gamma} \parallel \mathbf{w} \parallel\parallel \mathbf{e} \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2
\end{aligned}
$$

where M is a positive scalar constant satisfying $M \parallel \mathbf{x}_{td} - \mathbf{x}_t \parallel \geq \parallel \mathbf{G}(\mathbf{x}_{td}) - \mathbf{G}(\mathbf{x}_t) \parallel \forall \mathbf{x}_{td}, \mathbf{x}_t$ (the boundedness of the partial derivatives of $\mathbf{G}$ ensures that such an M exists). Define Q_c^* as

$$
Q_c^* = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma} & 0 & -\frac{M}{2\gamma} \\ 0 & (\gamma - \frac{k_2}{k_1\gamma})\lambda_m(H) & -\lambda_M(H) \\ -\frac{M}{2\gamma} & -\lambda_M(H) & k_1\gamma \end{bmatrix}
$$

(with Q_c^* positive-definite if γ is chosen large enough) and let $Q_c = \mathrm{diag}[\lambda_m(Q_c^*), k_a]$. Then the following bound on $\dot{V}_3$ in (23) can be obtained through routine manipulation:

$$
\dot{V}_3 \leq -(\lambda_m(Q_c) - \frac{\eta_{15}}{\gamma}{V_3}^{1/2}) \parallel \mathbf{z}_2 \parallel^2 \qquad (24)
$$

where η_{15} is a scalar constant which does not increase as γ is increased. Let $c_c(t) = \lambda_m(Q_c) - \eta_{15}{V_3}^{1/2}(t)/\gamma$ and choose γ large enough so that $c_c(0) > 0$ (this is always possible). Then (24) implies that $c_c(0) \leq c_c(t) \;\; \forall t$, so that $\dot{V}_3 \leq -c_c(0) \parallel \mathbf{z}_2 \parallel^2$. Standard arguments can then be used to show that all signals are bounded and that $\mathbf{z}_2$ converges to zero (and therefore that $\mathbf{e}$ converges to zero) [25]. Finally, note that the convergence of $\mathbf{e}$ to zero implies that $\mathbf{P}_t \rightarrow \mathbf{P}_{td}$, which in turn implies that $\mathbf{x}_t \rightarrow \mathbf{x}_{td}$. ■

4.2 Experimental Results

The performance of the proposed adaptive position/force control strategy is now studied in an experimental investigation. The experimental facility utilized for this study is the New Mexico State University Robotics Laboratory and consists of an IMI Zebra Zero robot arm and the associated control computer (see Figure 1). All control software is written in 'C' and is hosted on an IBM-compatible 486 personal computer. Note that the Zebra robot is one of the few commercially available manipulators with a force/torque sensor integrated directly into their design; this fact motivated the selection of this robot for both the simulation and experimental investigations presented in this paper.

The proposed adaptive position/force control scheme (17) is implemented in the present experimental study. The task specified in the experiment requires the manipulator to exert a user-defined contact force on the environment while tracking a position trajectory on the constraint surface. Two different environments are utilized in the experiments: a "soft" environment consisting of a large desk blotter placed on the laboratory table, and a "stiff"environment obtained by mounting a metal plate on the laboratory table. In each experiment, the reaction surface is placed in the robot workspace so that it is oriented normal to the z axis. The manipulator is commanded to exert a contact force of 10.0N on the reaction surface, maintain its initial end-effector orientation, and track the following desired position trajectory in the x, y plane: $x_d(t) = 1.0 + 0.5\sin\pi t/4$, $y_d(t) = -1.0 + 0.5(1 - \cos\pi t/4)$ for $t \in [0, 4]$. The desired position/force control task is therefore to exert a constant force normal to the reaction surface while tracking a semicircular trajectory on this surface.

The experiments are designed to evaluate the effectiveness of the proposed controller (17) for simultaneously controlling end-effector position and contact force, and to examine the sensitivity of the controller to a large variation in environmental stiffness. For all experiments, the controller gains are set to $k_1 = 10$, $k_2 = 20$, $\gamma = 5$, and $k_a = 100$. All of the adaptive elements are initialized to zero, and the adaptation gains are set as follows: $\beta_1 = 10$, $\beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 1$, and $\sigma_i = 0.1$ for $i = 1, 2, ..., 6$ (for all gains, the units for force and length are Newton and centimeter, and $\mathbf{e}$ is computed by "normalizing" the force error component through division by 1000). The adaptive controller is applied to the robot with a sampling period of seven milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of seven milliseconds. The results of these experiments are given in Figures 3a-3c, with Figure 3a depicting the position trajectory tracking performance for the stiff environment (the soft environment performance was virtually identical), Figure 3b showing the force control response for the soft environment, and Figure 3c illustrating the force control response for the stiff environment. It can be seen from the plots that the desired position trajectory is closely tracked and the desired force setpoint is accurately maintained, despite the lack of *a priori* information concerning the manipulator and the environment. Additionally, it is observed that the performance is robust to large variations in environmental characteristics (desk blotter and steel plate).

5. Conclusions

This paper presents two new adaptive compliant motion control schemes for uncertain RLED manipulators. It is shown that the schemes ensure semiglobal uniform boundedness of all signals, and that the ultimate size of the system errors can be made arbitrarily small. The capabilities of the proposed schemes are illustrated through both computer simulations and laboratory experiments with an IMI Zebra Zero manipulator. These studies indicate that the adaptive impedance controller provides an accurate and robust means of realizing the desired end-effector impedance characteristics, and that the adaptive position/force control scheme represents an effective method of controlling end-effector position and force independently. Future work will address the problem of controlling uncertain RLED robotic systems when they are subjected to more general classes of contraints. Progress in this direction for the special case of rigid manipulators performing waste management operations are reported in [26].

6. Acknowledgments

The research described in this paper was supported in part through contracts with the Army Research Office and Sandia National Laboratories.

7. References

1. Raibert, M. and J. Craig, "Hybrid Position/Force Control of Manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, No. 2, 1981, pp. 126-133

2. Hogan, N., "Impedance Control: An Approach to Manipulation, Parts I-III", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, No. 1, 1985, pp. 1-24

3. Good, M., L. Sweet, and K. Strobel, "Dynamic Models for Control System Design of Integrated Robot and Drive Systems", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, 1985, pp. 53-59

4. Eppinger, S. and W. Seering, "Introduction to Dynamic Models for Robot Force Control", *IEEE Control Systems Magazine*, Vol. 7, No. 2, 1987, pp. 48-52

5. Tarn, T., A. Bejczy, X. Yun, and Z. Li, "Effect of Motor Dynamics on Nonlinear Feedback Robot Arm Control", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 1, 1991, pp. 114-122

6. Taylor, D., "Composite Control of Direct-Drive Robots", *Proc. IEEE Conference on Decision and Control*, Tampa, FL, December 1989

7. Dawson, D., Z. Qu, J. Carrol, and M. Bridges, "Control of Robot Manipulators in the

Presence of Actuator Dynamics", *International Journal of Robotics and Automation*, Vol. 8, No. 1, 1993, pp. 13-21

8. Canudas de Wit, C., R. Ortega, and S. Seleme, "Robot Motion Control Using Induction Motor Drives", *Proc. IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 1993

9. Tarkianien, M. and Z. Shiller, "Time Optimal Motions of Manipulators With Actuator Dynamics", *Proc. IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 1993

10. Dawson, D., Z. Qu, and J. Carrol, "Tracking Control of Rigid-Link Electrically-Driven Robot Manipulators", *International Journal of Control*, Vol. 56, No. 5, 1992, pp. 991-1006

11. Mahmoud, M., "Robust Control of Robot Arms Including Motor Dynamics", *International Journal of Control*, Vol. 58, No. 4, 1993, pp. 853-873

12. Bridges, M., D. Dawson, and X. Gao, "Adaptive Control of Rigid-Link Electrically-Driven Robot Manipulators", *Proc. IEEE Conference on Decision and Control*, San Antonio, TX, December 1993

13. Guenther, R. and L. Hsu, "Variable Structure Adaptive Cascade Control of Rigid-Link Electrically-Driven Robot Manipulators", *Proc. IEEE Conference on Decision and Control*, San Antonio, TX, December 1993

14. Stepanenko, Y. and C. Su, "Adaptive Motion Control of Rigid-Link Electrically-Driven Robot Manipulators", *Proc. IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994

15. Bridges, M. and D. Dawson, "Adaptive Control of Rigid-Link Electrically-Driven Robots Actuated with Brushless DC Motors", *Proc. IEEE Conference on Decision and Control*, Orlando, FL, December 1994

16. Colbaugh, R., K. Glass, and E. Barany, "Adaptive Stabilization and Tracking Control of Electrically Driven Manipulators", *Journal of Robotic Systems*, Vol. 13, No. 3, 1996

17. Kwan, C., "Hybrid Force/Position Control for Manipulators with Motor Dynamics Using a Sliding-Adaptive Approach", *IEEE Transactions on Automatic Control*, Vol. 40, No. 5, 1995, pp. 963-968

18. Seraji, H., "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 4, 1989, pp. 472-490

19. Glass, K., R. Colbaugh, D. Lim, and H. Seraji, "Real-time Collision Avoidance for Redundant Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, 1995, pp. 448-457

20. Spong, M. and M. Vidyasagar, *Robot Dynamics and Control*, Wiley, New York, 1989

21. Corless, M., "Guaranteed Rates of Exponential Convergence for Uncertain Systems", *Journal of Optimization Theory and Applications*, Vol. 64, No. 3, 1990, pp. 481-494

22. Colbaugh, R., H. Seraji, and K. Glass, "Adaptive Compliant Motion Control for Dexterous Manipulators", *International Journal of Robotics Research*, Vol. 14, No. 3, 1995, pp. 270-280

23. Lozano, R. and B. Brogliato, "Adaptive Hybrid Force-Position Control for Redundant

Manipulators", *IEEE Transactions on Automatic Control*, Vol. 37, No. 10, 1992, pp. 1501-1505

24. Colbaugh, R., "Modeling Constrained Mechanical Systems for Advanced Controller Development", Robotics Laboratory Report 95-11, New Mexico State University, May 1995

25. Narendra, K. and A. Annaswamy, *Stable Adaptive Systems*, Prentice Hall, Englewood Cliffs, NJ, 1989

26. Colbaugh, R., E. Barany, and K. Glass, "Adaptive Control of Constrained Robotic Systems for Waste Management Applications", *International Journal of Environmentally Conscious Design and Manufacturing*, Vol. 5, 1996 (in press)

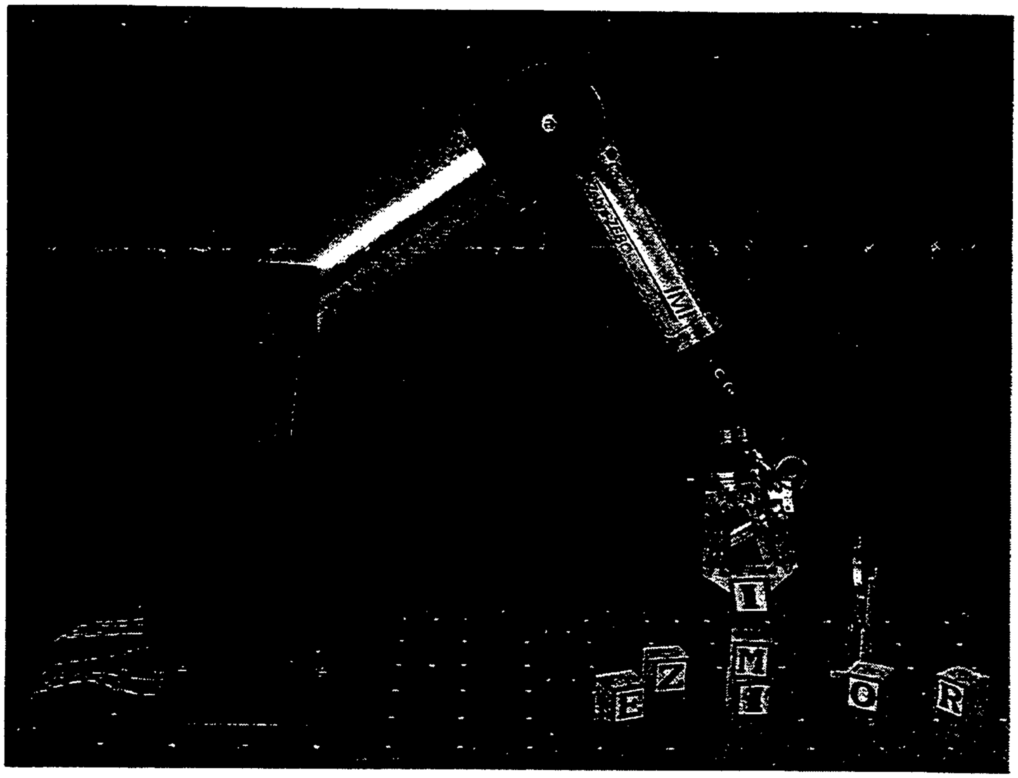


Figure 1: Illustration of Zebra Zero manipulator

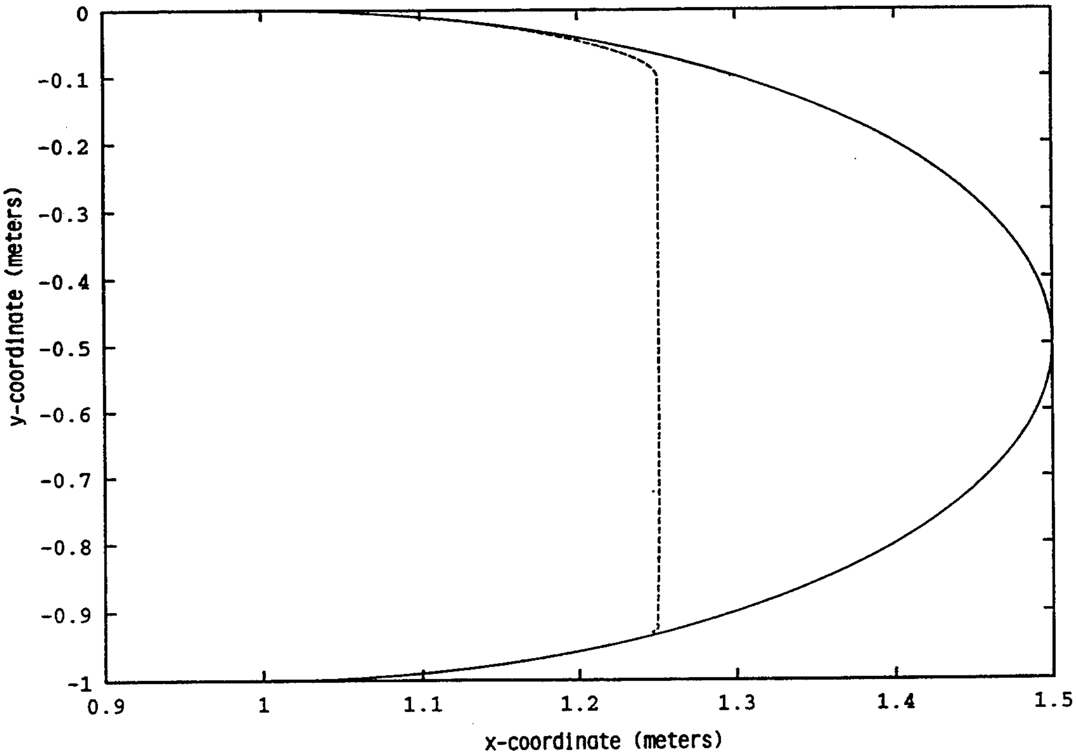


Figure 2a: Reference (solid) and actual (dashed) end-effector trajectory in x, y plane in impedance control simulation example.

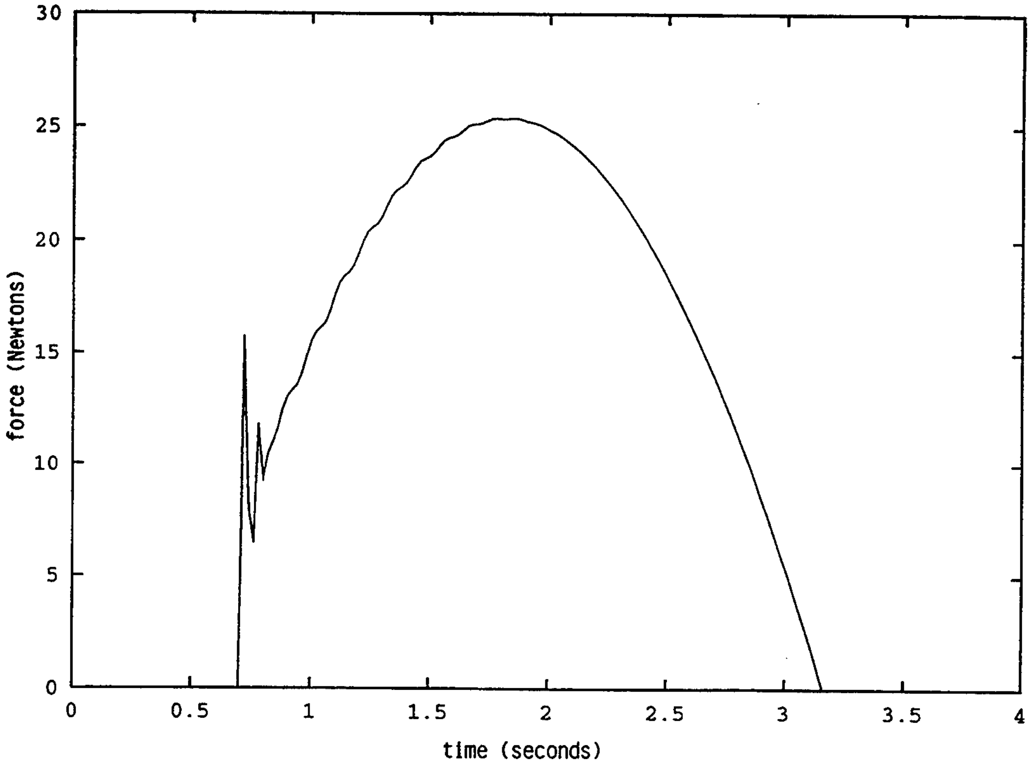


Figure 2b: End-Effector/environment contact force in impedance control simulation example.

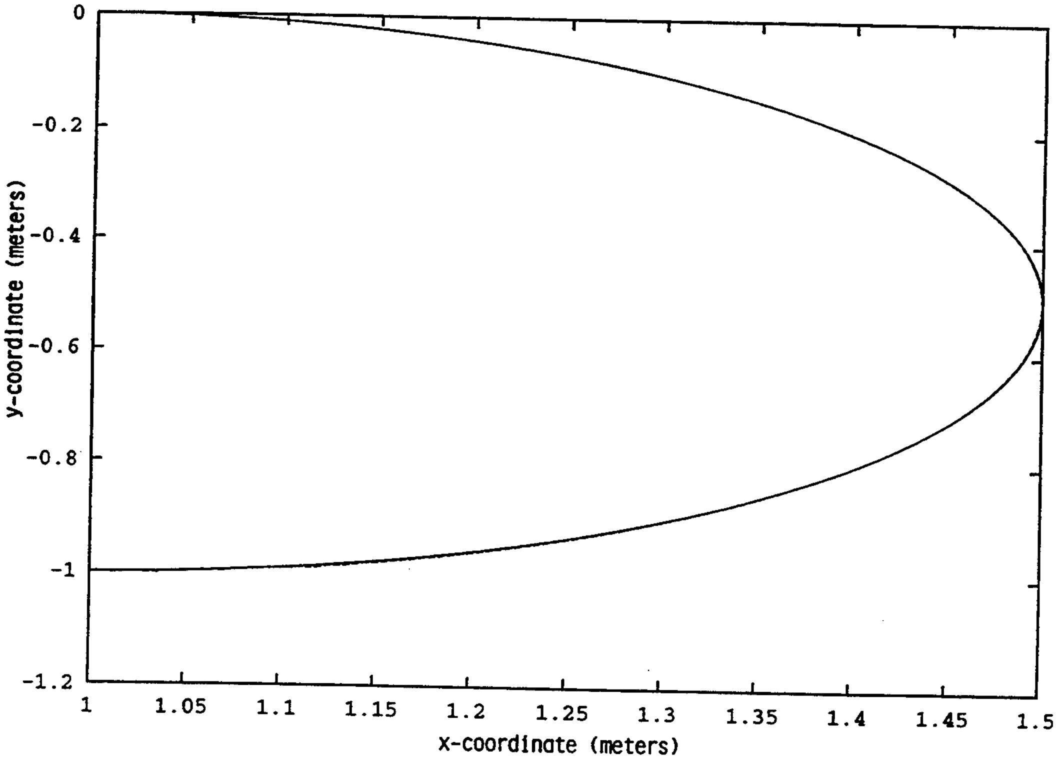


Figure 3a: Desired (solid) and actual (dashed) end-effector trajectory in x, y plane in position/force control experiment.

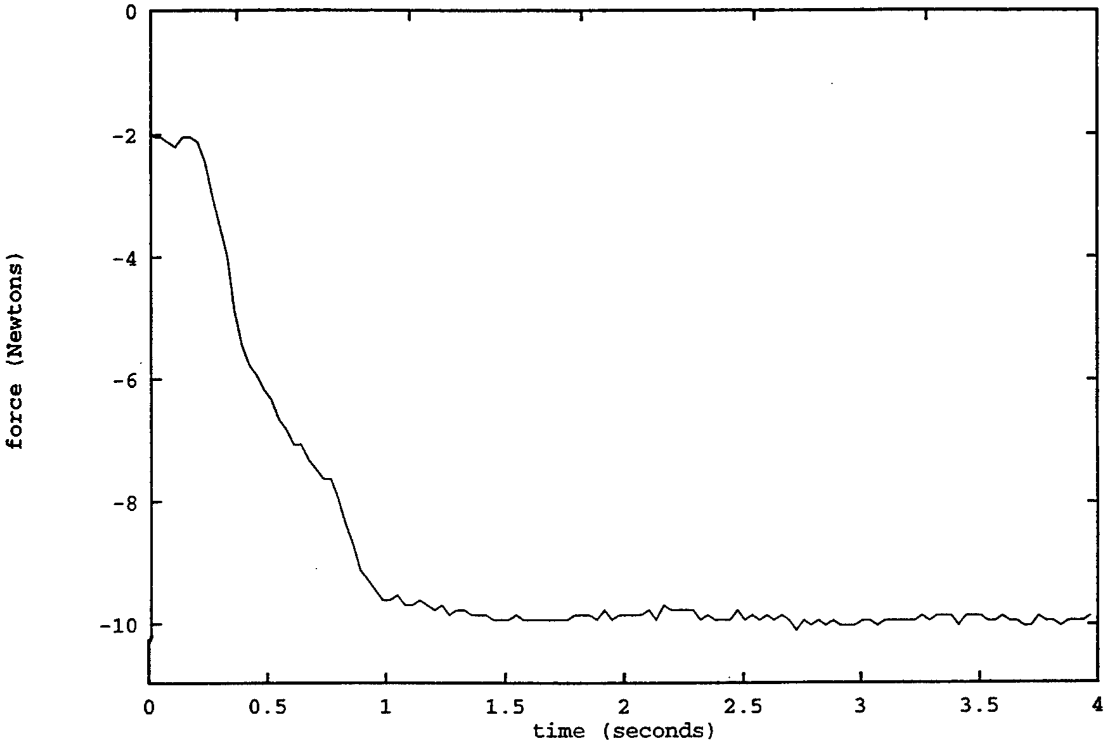


Figure 3b: End-Effector/environment contact force in position/force control experiment (soft environment).

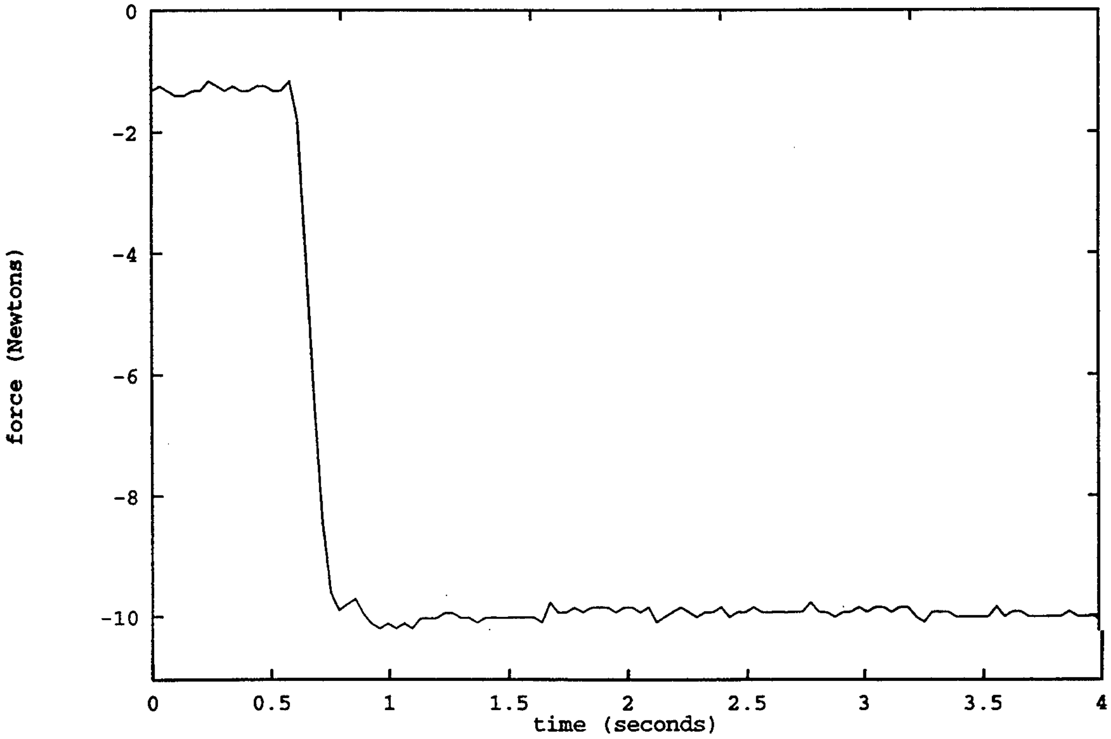


Figure 3c: End-Effector/environment contact force in position/force control experiment (stiff environment).

to appear: International Journal of Robotics Research

ADAPTIVE REGULATION OF MANIPULATORS USING ONLY POSITION MEASUREMENTS

R. Colbaugh* K. Glass* E. Barany†

Departments of *Mechanical Engineering and †Mathematical Sciences
New Mexico State University, Las Cruces, NM 88003 USA

Abstract

This paper considers the motion control problem for uncertain robot manipulators for the case in which only joint position measurements are available, and proposes an adaptive controller as a solution to this problem. The proposed control strategy is very general and computationally efficient, requires very little information regarding the manipulator model or the payload, and ensures that the position error possesses desirable convergence properties: semiglobal asymptotic convergence if no external disturbances are present and semiglobal convergence to an arbitrarily small neighborhood of zero in the presence of bounded disturbances. It is shown that the adaptive controller can be modified to provide accurate tracking control through the introduction of additional feedforward elements in the control law. Computer simulation results are given for a PUMA 560 robot and a Zebra Zero manipulator; these results demonstrate that accurate and robust position regulation and trajectory tracking can be achieved by using the proposed controller. Experimental results are presented for a Zebra Zero manipulator and confirm that the proposed approach provides a simple and effective means of obtaining high performance motion control.

1. Introduction

The position regulation problem for robot manipulators involves designing a control strategy to cause the manipulator to evolve from its initial state to some desired final state, where typically the final velocity is to be zero. This is also referred to as the position control or point-to-point motion control problem, emphasizing the fact that only the final position is specified and the intermediate trajectory is not stipulated. It is well-known that simple proportional-derivative (PD) feedback controllers are capable of globally asymptotically stable regulation of rigid robots, provided that the effects of gravity on the manipulator are compensated. Ordinarily the requisite gravity compensation is achieved by including a gravity model in the control scheme, so that the resulting controller possesses a PD-plus-gravity structure [1].

While the PD-plus-gravity position control law is simple, elegant, and intuitively appealing, there are two potential difficulties associated with this approach. First, providing gravity compensation by including a model of the manipulator gravity torques in the control law can be undesirable because this requires precise *a priori* knowledge of both the structure and the parameter values for this model, including the effects of any payload. Particularly restrictive is the need for information concerning the payload, because in typical tasks many different payloads are encountered and it is unrealistic to assume that the properties of all payloads are accurately known. Additionally, this requirement limits the modularity and portability of the controllers, so that implementation of these schemes with a new manipulator can involve considerable effort. The second drawback with PD-plus-gravity regulation schemes is that these controllers include a linear state feedback component and therefore require full state measurement, which can be problematic in practice. For example, although manipulators are generally equipped with high-precision sensors capable of accurately measuring joint position, determining the joint velocities is more challenging and typically the velocity measurements are contaminated with noise [2]. In fact, velocity sensors are frequently omitted altogether in the interest of saving money, volume, and weight in manipulator construction [2], so that in many applications only position information is available for feedback.

Recognizing these difficulties, several researchers have studied methods of addressing one or the other problem. Studies focusing on reducing the information required concerning the manipulator gravity model have tended to concentrate on adaptive control approaches. For instance, the schemes presented in [3-5] possess a PD-plus-gravity structure in which the (constant) inertial parameters for the manipulator and payload are assumed to be unknown, and this uncertainty is compensated adaptively. Implementation of these adaptive controllers still requires knowledge of the structure of the gravity model and measurements of joint velocity, however. Techniques for eliminating the need for velocity measurements in manipulator controllers have been proposed in [e.g., 6-10]. One approach to this problem is to design an observer for estimating the velocity utilizing the available (high-quality) position information, and then to use this estimated velocity in place of the actual velocity in the state feedback component of the control law; this strategy is used in [6-8] to solve the manipulator tracking problem. Alternatively, the simple structure of the PD-plus-gravity solution to the manipulator regulation problem suggests a correspondingly simple solution to the regulation problem when only position measurements are available. Thus elegant control laws composed of a gravity compensation term and a linear dynamic (first order) compensator have been proposed very recently in [9,10]. It is noted that while these schemes eliminate the need for velocity measurements, they still require knowledge of the gravity model for the manipulator and payload to provide accurate position control.

Research in which the above objectives are combined, so that accurate position regulation is achieved without gravity model information or velocity measurements, appears to be lacking; preliminary results along these lines are reported by the authors in [11]. An additional observation concerning this work is that none of the regulation schemes presented in [3-10] have been verified through a comprehensive simulation and experimental study involving industrial robots.

This paper introduces an adaptive position regulation scheme that does not require joint velocity measurements or knowledge of the gravity model for the manipulator or payload. The proposed controller is general, computationally simple, and easy to implement. It is shown that the adaptive strategy ensures semiglobal uniform boundedness of all signals in the presence of bounded external disturbances and possesses desirable convergence properties: asymptotic convergence of the position error if no disturbances are present and convergence of the error to an arbitrarily small neighborhood of zero in the presence of disturbances. Moreover, it is demonstrated that the regulation scheme can be modified to provide accurate tracking control through the introduction of additional feedforward elements in the control law. The results of extensive computer simulations and hardware experiments are presented to illustrate the efficacy of the proposed approach.

The paper is organized as follows. In Section 2 some preliminary facts of relevance to the manipulator motion control problem are established. The adaptive regulation scheme is presented and analyzed in Section 3. The performance of the controller is illustrated in Section 4 through a computer simulation study and in Section 5 through an experimental investigation. Finally, Section 6 summarizes the paper and draws some conclusions.

2. Preliminaries

This paper considers the adaptive position regulation problem for uncertain manipulators using only position measurements. Let $\mathbf{y}$ define the position and orientation of the robot end-effector relative to a fixed user-defined reference frame and note that, in the most general case, the elements of $\mathbf{y}$ are local coordinates for some smooth manifold $\mathcal{M}$ of dimension m. The forward kinematic and differential kinematic maps between the robot joint coordinates $\theta \in \mathcal{N}$ and the end-effector coordinates $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{h}(\theta), \qquad \dot{\mathbf{y}} = J(\theta)\dot{\theta} \tag{1}$$

where $\mathcal{N}$ is a smooth manifold of dimension n, $\mathbf{h} : \mathcal{N} \rightarrow \mathcal{M}$, and $J \in \Re^{m \times n}$ is the end-effector Jacobian matrix.

It is often desirable to formulate the manipulator control problem in terms of generalized coordinates other than the joint coordinates θ. If the manipulator is nonredundant (so that $m = n$) and is in a region of the workspace where J has full rank, then the elements of $\mathbf{y}$ are also generalized coordinates for the manipulator. Observe that the end-effector coordinates $\mathbf{y}$ are typically more task-relevant than the joint coordinates θ, so that developing the controller in terms of $\mathbf{y}$ can lead to improved performance, efficiency, and implementability. A task-space formulation can also be realized if the manipulator is kinematically redundant (so that $m < n$) by utilizing, for example, the *configuration control* approach [e.g., 12,13]. In what follows, we shall consider nonredundant and redundant robots together and formulate the manipulator control problem in terms of a set of n generalized coordinates $\mathbf{x}$. To retain generality, we shall require only that the elements of $\mathbf{x}$ are local coordinates for some smooth manifold $\mathcal{X}$ of dimension n, and that the kinematic relationship between θ and $\mathbf{x}$ is known and can be written in a form analogous to (1):

$$\mathbf{x} = \mathbf{h}_a(\theta), \qquad \dot{\mathbf{x}} = J_a(\theta)\dot{\theta} \tag{2}$$

where $\mathbf{h}_a : \mathcal{N} \rightarrow \mathcal{X}$ and $J_a \in \Re^{n \times n}$. Observe that for $\mathbf{x}$ to be a valid generalized coordinate vector the elements of $\mathbf{x}$ must be independent in the region of interest; thus it will be assumed in our development that J_a is of full rank. Note also that joint-space control is trivially recovered by defining $\mathbf{h}_a$ to be the identity map.

Consider the manipulator dynamic model written in terms of the generalized coordinates $\mathbf{x}$:

$$\mathbf{F} = H(\mathbf{x})\ddot{\mathbf{x}} + V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}(\mathbf{x}) + \mathbf{d}(t) \tag{3}$$

where $\mathbf{F} \in \Re^n$ is the generalized force associated with $\mathbf{x}$, $H \in \Re^{n \times n}$ is the manipulator inertia matrix, $V_{cc} \in \Re^{n \times n}$ quantifies Coriolis and centripetal acceleration effects, and $\mathbf{G} \in \Re^n$ is the vector of gravity forces. The term $\mathbf{d} \in \Re^n$ is an unknown vector, assumed to be bounded with bounded first derivative with respect to time, that can represent unmodelled effects or external disturbances; this term is included in the system dynamics to permit the robustness of the proposed regulation scheme to be studied. It is well-known that the dynamics (3) possesses considerable structure. For example, for any set of generalized coordinates $\mathbf{x}$, the matrix H is bounded, symmetric and positive-definite, the matrices $\dot{H}$ and V_{cc} are bounded in $\mathbf{x}$, depend linearly on $\dot{\mathbf{x}}$, and are related according to $\dot{H} = V_{cc} + V_{cc}^T$. The vector of gravity forces $\mathbf{G}$ is the gradient of a potential energy function and is bounded with bounded first partial derivatives. Additionally, $V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\mathbf{y} = V_{cc}(\mathbf{x}, \mathbf{y})\dot{\mathbf{x}} \ \ \forall \mathbf{y}$, and if $\mathbf{y}$ and $\dot{\mathbf{y}}$ are bounded then $V_{cc}(\mathbf{x}, \mathbf{y})$ is bounded and $\dot{V}_{cc}(\mathbf{x}, \mathbf{y})$ grows linearly with $\dot{\mathbf{x}}$. All of these properties except the last one are well-known [e.g., 14]; the last property of the Coriolis/centripetal matrix V_{cc} is essentially given in [15] and is easy to see by expanding V_{cc} in terms of Christoffel symbols.

The control system proposed in this paper consists of two subsystems: an adaptive scheme that provides the control input $\mathbf{F}$ required to ensure that the system (3) evolves from its initial state to the desired final state $\mathbf{x} = \mathbf{x}_d$, $\dot{\mathbf{x}} = \mathbf{0}$, and an algorithm for mapping the control input $\mathbf{F}$ to a physically realizable joint-space control torque $\mathbf{T} \in \Re^n$. If $\mathbf{x}$ is chosen as described above then the required mapping between $\mathbf{F}$ and $\mathbf{T}$ is simply $\mathbf{T} = {J_a}^T\mathbf{F}$. Observe that the desired task-space control input $\mathbf{F}$ can always be achieved in this way because it has been assumed that J_a is of full rank (this is the standard assumption in task-space controller development). It is noted that this rank assumption could be violated if, during the robot's motion, a kinematic singularity is encountered; in this event J_a loses rank and the desired control input $\mathbf{F}$ may or may not be realizable. However, extensive computer simulations and hardware experiments have indicated that task-space implementations of position regulation schemes are quite robust to kinematic singularities, and indeed this robustness is to be expected. For example, under position regulation the manipulator evolves to the goal position using the so called "natural motion" of the system, which provides considerable robustness to singularities [e.g., 16,17]. Additionally, a direct task-space implementation eliminates the need for an inverse kinematic map and instead achieves the necessary transformation from task-space to joint-space through a mapping of generalized forces (i.e., by mapping $\mathbf{F}$ to $\mathbf{T}$). This strategy has been shown both theoretically and in practice to provide significant robustness to singularities [e.g., 12,13,17-20]. Thus, for the remainder of the paper, we make the usual task-space control assumption that the control input $\mathbf{F}$ can be commanded as desired (through the equivalent joint torque vector $\mathbf{T} = {J_a}^T\mathbf{F}$), and we focus in our subsequent discussion on the adaptive regulation of the system (3).

3. Adaptive Motion Control

We now turn to the development of an adaptive strategy for position regulation of uncertain manipulators which requires only position measurements. Before presenting our main results, we briefly consider the nonadaptive case to fix some ideas and introduce a useful Lyapunov function.

3.1 Nonadaptive Case

In [9], Berghuis and Nijmeijer consider the problem of robot regulation using only position measurements and propose the following elegant solution based on the seminal work of Takegaki and Arimoto [1]:

$$\begin{aligned} \mathbf{F} &= \mathbf{G}(\mathbf{x}) + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \qquad (4) \\ \dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \end{aligned}$$

where $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ is the position regulation error (recall that $\mathbf{x}_d$ is constant), k_1, k_2, γ are positive scalar constants, and where we have reformulated the scheme given in [9] slightly to make our subsequent analysis more convenient. Observe that (4) is implementable without velocity information because, although $\dot{\mathbf{w}}$ depends on $\dot{\mathbf{e}}$, the control law requires only $\mathbf{w}$ and this term depends only on $\mathbf{e}$ (as can be verified through direct integration of the $\dot{\mathbf{w}}$ equation). The scheme (4) can be viewed as a natural extension of the well-known PD-plus-gravity control law proposed in [1], with the velocity feedback component of the controller in [1] replaced by a first order linear compensator. In fact, it will be seen below that $\mathbf{w}$ provides a simple and effective means of injecting damping into the closed-loop system without using velocity feedback. The reader interested in additional background, motivation, and implementation details concerning the position regulation scheme (4) is referred to the paper [9].

It is shown in [9] that (4) globally stabilizes the *nominal* manipulator dynamics (3) (i.e., the dynamics (3) with $\mathbf{d} = \mathbf{0}$) at the equilibrium $\mathbf{e} = \dot{\mathbf{e}} = \mathbf{w} = \mathbf{0}$; however, the stability analysis in [9] employs a Lyapunov function with a negative-semidefinite derivative, so that it is not clear how to extend this analysis to the adaptive case. In what follows, we present an alternative stability analysis which demonstrates that the equilibrium point is exponentially stable, and which forms the basis for our derivation of an adaptive solution to the problem.

Lemma 1: The controller (4) exponentially stabilizes the *nominal* manipulator (3) (with $\mathbf{d} = \mathbf{0}$) at the equilibrium $\mathbf{e} = \dot{\mathbf{e}} = \mathbf{w} = \mathbf{0}$ provided γ is chosen sufficiently large.

Proof: Applying the controller (4) to the nominal manipulator dynamics (3) yields the closed-loop system

$$\begin{aligned} H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} &= 0 \\ \dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \end{aligned} \tag{5}$$

Consider the Lyapunov function candidate

$$V_1 = \frac{1}{2}\dot{\mathbf{e}}^T H\dot{\mathbf{e}} + \frac{1}{2}k_2\gamma^2\mathbf{e}^T\mathbf{e} + \frac{1}{2}k_1\mathbf{w}^T\mathbf{w} + \frac{k_2}{k_1\gamma}\mathbf{e}^T H\dot{\mathbf{e}} - \frac{1}{\gamma}\mathbf{w}^T H\dot{\mathbf{e}} \tag{6}$$

and note that V_1 in (6) is positive-definite and radially-unbounded provided γ is chosen large enough. Differentiating (6) along (5) and simplifying using the structural properties of the manipulator dynamics summarized in Section 2 yields

$$\dot{V}_1 = -(\gamma - \frac{k_2}{k_1\gamma})\dot{\mathbf{e}}^T H\dot{\mathbf{e}} - \frac{k_2^2\gamma}{k_1}2\|\text{-e0}\|^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 + 2\mathbf{w}^T H\dot{\mathbf{e}}$$

$$
\begin{aligned}
&+\frac{1}{\gamma}\dot{\mathbf{e}}^T[\frac{k_2}{k_1}V_{cc}\mathbf{e}-V_{cc}\mathbf{w}] \\
&\leq -(\gamma-\frac{k_2}{k_1\gamma})\lambda_{min}(H)\parallel\dot{\mathbf{e}}\parallel^2-\frac{k_2^2\gamma}{k_1}\parallel\mathbf{e}\parallel^2-k_1\gamma\parallel\mathbf{w}\parallel^2 \\
&+2\lambda_{max}(H)\parallel\mathbf{w}\parallel\parallel\dot{\mathbf{e}}\parallel+\frac{k_2k_{cc}}{k_1\gamma}\parallel\mathbf{e}\parallel\parallel\dot{\mathbf{e}}\parallel^2+\frac{k_{cc}}{\gamma}\parallel\mathbf{w}\parallel\parallel\dot{\mathbf{e}}\parallel^2 \qquad (7)
\end{aligned}
$$

where $\lambda_{min}(\cdot),\lambda_{max}(\cdot)$ denote the minimum and maximum eigenvalue of the matrix argument, respectively, and k_{cc} is a scalar upper bound on the linear dependency of V_{cc} on $\dot{\mathbf{x}}$ (i.e., $\parallel V_{cc}\parallel_F\leq k_{cc}\parallel\dot{\mathbf{x}}\parallel\ \forall\mathbf{x}$ with $\parallel\cdot\parallel_F$ the Frobenius matrix norm). Define $\mathbf{z}=[\parallel\mathbf{e}\parallel\ \ \parallel\dot{\mathbf{e}}\parallel\ \ \parallel\mathbf{w}\parallel]^T$ and

$$
Q_1=\begin{bmatrix}\frac{k_2^2\gamma}{k_1} & 0 & 0\\ 0 & (\gamma-\frac{k_2}{k_1\gamma})\lambda_{min}(H) & -\lambda_{max}(H)\\ 0 & -\lambda_{max}(H) & k_1\gamma\end{bmatrix}
$$

and note that Q_1 is positive-definite if γ is chosen sufficiently large. The following bound on $\dot{V}_1$ in (7) can then be established:

$$
\begin{aligned}
\dot{V}_1 &\leq -\mathbf{z}^TQ_1\mathbf{z}+\frac{k_2k_{cc}}{k_1\gamma}\parallel\mathbf{e}\parallel\parallel\dot{\mathbf{e}}\parallel^2+\frac{k_{cc}}{\gamma}\parallel\mathbf{w}\parallel\parallel\dot{\mathbf{e}}\parallel^2 \\
&\leq -(\lambda_{min}(Q_1)-\frac{\alpha_1}{\gamma}{V_1}^{1/2})\parallel\mathbf{z}\parallel^2 \qquad (8)
\end{aligned}
$$

for some scalar constant α_1 which does not increase as γ is increased. Let $c_1(t)=\lambda_{min}(Q_1)-\alpha_1{V_1}^{1/2}(t)/\gamma$ and choose γ large enough so that $c_1(0)>0$ (this is always possible). The inequality (8) then implies that $c_1(0)\leq c_1(t)\ \ \forall t$, so that $\dot{V}_1\leq -c_1(0)\parallel\mathbf{z}\parallel^2$. Standard arguments can then be used to show that $\mathbf{e},\dot{\mathbf{e}},\mathbf{w}$ converge exponentially to zero [e.g., 21]. □

Observe that in the preceding stability analysis a negative-definite Lyapunov function derivative has been obtained. It will be shown in what follows that, as a consequence, this analysis can be readily extended to the adaptive case. Note also that the stability properties established in Lemma 1 are *semiglobal* in the sense that the region of attraction can be increased arbitrarily by increasing the controller gain γ. It is stressed that this does not imply that γ must be chosen to be overly large; indeed, we have found that excellent performance is obtainable with this gain set to quite modest values.

3.2 Adaptive Regulation Scheme

As indicated above, it is desirable to eliminate the need for information concerning the manipulator gravity model in position regulation strategies. Consider the following adaptive approach to realizing this objective:

$$\begin{aligned}
\mathbf{F} &= \mathbf{f}(t) + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \\
\dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \\
\dot{\mathbf{f}} &= -\sigma\mathbf{f} + \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w})
\end{aligned} \tag{9}$$

where $\mathbf{f}(t) \in \Re^n$ is the adaptive element, β is a positive scalar constant, and σ is defined as follows: $\sigma = 0$ if $\| \mathbf{f} \| \leq f_{max}$ and $\sigma = \sigma_0$ otherwise, with σ_0 and f_{max} positive scalars. The adaptive element $\mathbf{f}$ is intended to compensate for gravity torques $\mathbf{G}$ and external disturbances $\mathbf{d}$, β is the adaptation gain for $\mathbf{f}$, and σ is defined to provide a "switching σ-modification" adaptation law [22]. This approach to adaptive control has been proposed for linear systems to provide robust performance in the presence of external disturbances while retaining asymptotic convergence of the system error in the ideal case in which no disturbances are present [22]. In what follows, it will be shown that the formulation (9) yields a similar robust adaptive performance for the problem of regulating the nonlinear manipulator system (3). Observe that (9) is implementable without velocity information because, although $\dot{\mathbf{w}}$ and $\dot{\mathbf{f}}$ depend on $\dot{\mathbf{e}}$, the control law terms $\mathbf{w}$ and $\mathbf{f}$ depend only on $\mathbf{e}$ (as can be verified through direct integration of the $\dot{\mathbf{w}}$ and $\dot{\mathbf{f}}$ equations).

The stability properties of the proposed adaptive regulation strategy (9) are summarized in the following theorem.

Theorem 1: If γ is chosen sufficiently large then the adaptive scheme (9) ensures that (3) evolves with all signals (semiglobally) uniformly bounded and so that:

1. In the presence of external disturbances, $\mathbf{e}$ and $\dot{\mathbf{e}}$ converge asymptotically to a compact set which can be made arbitrarily small (by increasing β).
2. In the absence of external disturbances, $\mathbf{e}$ and $\dot{\mathbf{e}}$ converge asymptotically to zero.

Proof: The proof is given in Appendix A. □

Several observations can be made concerning the adaptive position regulation strategy (9). First, the stability analysis presented in Appendix A demonstrates that this scheme ensures that the manipulator dynamics (3) evolves from any initial state to any desired final configuration with uniform boundedness of all signals and with desirable convergence properties: asymptotic convergence to zero if no external disturbances are present and convergence to an arbitrarily small neighborhood of zero in the presence of bounded disturbances. Note that, in the presence of disturbances, the ultimate size of the regulation error can be reduced as desired simply by increasing a single controller parameter (the adaptation gain β). The controller is very simple, utilizes only measurements of joint position, and does not require knowledge of the structure or parameter values for the manipulator model or the payload. Thus the proposed scheme provides a computationally efficient, modular, readily implementable, and robust approach to manipulator position regulation in task-space or joint-space. An additional observation is that the scalar controller gains k_1, k_2, γ, and β can be replaced with the appropriate diagonal matrices for increased flexibility and performance; scalar gains are used above for simplicity of development.

Finally, it is noted that robust and accurate position regulation with good transient performance is obtainable with a wide range of values for the controller parameters k_1, k_2, γ, β, σ_0, and f_{max}. This may be inferred from an examination of the stability proof in Appendix A, where the role of these parameters in the closed-loop system evolution is made explicit, and is clearly demonstrated in the computer simulations and hardware experiments presented later in the paper. As a consequence, we have found that it is easy to choose values for these parameters which provide good performance for a given robot manipulator; indeed, we have found that a few iterations is sufficient to obtain such parameter values. Of course, it is useful to present some guidelines for use in selecting these controller parameters. Such guidelines may be obtained from a study of the stability proof in Appendix A and an inspection of the controller structure in (9). For example, reasonable values for k_1, k_2, and γ can be obtained by setting $k_2 = 2k_1$ and choosing γ so that $k_2\gamma^2$ and $k_1\gamma^2$ are comparable to standard "PD" gains for the manipulator, respectively (there are well-known techniques for obtaining such values). The controller parameter f_{max} should be chosen so that $\| \mathbf{G}(\mathbf{x}) + \mathbf{d}(t) \| \leq f_{max}$; this is easily achieved since f_{max} can be chosen so that this inequality is satisfied for a very conservative upper bound on $\| \mathbf{G}(\mathbf{x}) + \mathbf{d}(t) \|$. Finally, the adaptation gain σ_0 should be chosen relatively small (compared with β), while the adaptation gain β can be increased until the desired positioning accuracy is obtained. Of course, β should not be set to an excessively large value since this can cause difficulties in digital implementation of the adaptive law. Additional details concerning the selection of the controller parameters are provided in the Sections 4 and 5, where a description of the results of the computer simulations and hardware experiments is given.

3.3 Extension to Tracking Control

It is interesting to note that the proposed approach to controller development is readily applicable to the robot trajectory tracking control problem, where the objective is to ensure that the manipulator (3) evolves from its initial state to the desired final state along some specified trajectory $\mathbf{x}_d(t) \in \mathcal{X}$ (where $\mathbf{x}_d$ is bounded with bounded derivatives). Again, our goal is to obtain a control scheme which does not require joint velocity measurements or knowledge of the manipulator dynamic model for implementation. Consider the adaptive tracking controller obtained from (9) through the introduction of feedforward elements in the control law:

$$\begin{aligned}\mathbf{F} &= A(t)\ddot{\mathbf{x}}_d + B(t)\dot{\mathbf{x}}_d + \mathbf{f}(t) + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \\ \dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}}\end{aligned} \tag{10}$$

where $A(t) \in \Re^{n\times n}$, $B(t) \in \Re^{n\times n}$ are (feedforward) adaptive gains which are adjusted according to the following simple update laws (the adaptation law for $\mathbf{f}(t)$ is repeated here for convenience of reference):

$$\begin{aligned}\dot{\mathbf{f}} &= -\sigma_1\mathbf{f} + \beta_1\mathbf{q} \\ \dot{A} &= -\sigma_2 A + \beta_2\mathbf{q}\ddot{\mathbf{x}}_d^T \\ \dot{B} &= -\sigma_3 B + \beta_3\mathbf{q}\dot{\mathbf{x}}_d^T\end{aligned} \tag{11}$$

where $\mathbf{q} = \dot{\mathbf{e}} + k_2\mathbf{e}/k_1\gamma - \mathbf{w}/\gamma$ represents a weighted and filtered error term and the σ_i and β_i are positive scalar adaptation gains. The feedforward terms $A(t)\ddot{\mathbf{x}}_d$ and $B(t)\dot{\mathbf{x}}_d$ which have been added to the regulation scheme (9) to obtain the tracking controller (10) are intended to compensate for the dynamic model terms $H\ddot{\mathbf{x}}$ and $V_{cc}\dot{\mathbf{x}}$, respectively; this compensation is necessary if accurate tracking control is to be achieved in an efficient manner. Observe that the scheme (10),(11) is implementable without velocity information because, although $\dot{\mathbf{w}}$, $\dot{\mathbf{f}}$, $\dot{A}$, and $\dot{B}$ depend on $\dot{\mathbf{e}}$, the control law requires the terms $\mathbf{w}$, $\mathbf{f}$, A, and B and these can be integrated so as to depend only on $\mathbf{e}$. †

The stability properties of the proposed adaptive tracking controller (10),(11) are summarized in the following theorem.

Theorem 2: The adaptive controller (10),(11) ensures that $\mathbf{e}, \dot{\mathbf{e}}, \mathbf{w}, \mathbf{f}, A, B$ are semiglobally uniformly bounded provided γ is chosen sufficiently large. Moreover, the state trajectory tracking error $\mathbf{e}, \dot{\mathbf{e}}$ is guaranteed to converge exponentially to a compact set which can be made arbitrarily small.

Proof: The proof is given in Appendix B. □

Several observations can be made concerning the adaptive trajectory tracking strategy (10),(11). First notice that, although the stability analysis presented in Appendix B is

† To see this in the case of A and B, note that terms of the form $\dot{\mathbf{e}}\ddot{\mathbf{x}}_d^T$ and $\dot{\mathbf{e}}\dot{\mathbf{x}}_d^T$ can be integrated as follows:

$$\int_0^t \dot{\mathbf{e}}\dot{\mathbf{x}}_d^T dt = \mathbf{e}\dot{\mathbf{x}}_d^T - \int_0^t \mathbf{e}\ddot{\mathbf{x}}_d^T dt$$

which eliminates the dependence on $\dot{\mathbf{e}}$.

somewhat complicated and involves complex expressions, the proposed control law is very simple, utilizes only measurements of joint position, and requires virtually no information concerning the manipulator dynamics. Thus the proposed control scheme provides a computationally efficient, modular, and readily implementable approach to manipulator trajectory tracking in task-space or joint-space. It is shown that the controller ensures uniform boundedness of all signals in the presence of bounded disturbances, and that the ultimate bounds on the trajectory tracking errors can be reduced arbitrarily by increasing the adaptation gains β_i. Notice also that exponential convergence ensures that the transient behavior of the tracking errors will be well-behaved. An additional observation is that the scalar controller gains k_1, k_2, γ and the adaptation gains σ_i and β_i can be replaced with the appropriate diagonal matrices for increased flexibility and performance; scalar gains are used above for simplicity of development. Note that, as was the case with the position regulation strategy (9), robust and accurate trajectory tracking is obtainable using (10),(11) with a wide range of values for the controller parameters. As a consequence, it is easy to obtain parameter values which provide good performance for a given robot manipulator, and it is mentioned that the guidelines provided for selecting controller parameters for the adaptive regulation scheme (9) are applicable here as well.

4. Simulation Results

The proposed adaptive position regulation strategy (9) and trajectory tracking scheme (10),(11) are now applied to two industrial robots through computer simulation. The robots chosen for this simulation study are the six degree-of-freedom (DOF) PUMA 560 manipulator and the six DOF IMI Zebra Zero robot. The simulation environment incorporates models of all important dynamic subsystems and phenomena, such as the full nonlinear arm dynamics, joint stiction, sensor noise, and transmission effects, and we therefore believe that the environment provides the basis for a realistic evaluation of controller performance [23]. One indication that this simulation environment does, indeed, provide the desired realism is the close agreement between the simulated response of the Zebra manipulator and the actual response of this robot observed in the hardware experiments presented in the following Section Additional verification of the fidelity of the simulation environment is presented in [11,13,20,23]. For all simulations the control laws are applied to the manipulator model with a sampling period of two milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of two milliseconds. It is noted that this simple method of numerical integration is commonly used in digital implementation of adaptive control laws and has proven to be quite effective for these applications. Both the PUMA 560 and Zebra

Zero manipulators possess conventional designs with six revolute joints configured in a "waist-shoulder-elbow-wrist" arrangement (see Figure 1). However, the two manipulators are very different in size: the PUMA 560 robot is a conventional industrial arm while the Zebra Zero robot is a much smaller laboratory arm (for example, the total reach of the Zebra Zero is less than 30in). It is noted that throughout the computer simulation study, the unit of length is inch, the unit of time is second, the unit of angle is degree, and the unit of force is pound, unless stated otherwise.

The first set of simulations demonstrates the capability of the proposed adaptive regulation scheme (9) to provide accurate task-space position regulation. In the first simulation, the PUMA robot is initially at rest with joint-space position $\theta(0) = [0, 90, 180, 0, -90, 0]^T$. The robot is commanded to move its end-effector 10in in the x-direction, 30in in the y-direction, and 20in in the z-direction, where all motions are specified relative to the base coordinate frame. Additionally, the robot is required to change its end-effector orientation from an initial orientation matrix R_{init} to a final orientation matrix R_{final}, where

$$R_{init} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad R_{final} = \begin{bmatrix} \sqrt{3}/2 & -1/2 & 0 \\ 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and where each rotation matrix R is specified relative to the base frame of the robot [14]. Note that, alternatively, the required orientation change can be specified using the familiar "equivalent-angle-axis" parameterization of orientation given by $\psi\hat{\mathbf{k}} \in \Re^3$, where $\hat{\mathbf{k}}$ is a unit vector aligned with the axis of rotation from the initial to the final desired orientation and ψ is the angle between these orientations measured about this axis; using this representation the desired orientation change is $\psi_d = 150°$ and $\hat{\mathbf{k}}_d = [0, 0, 1]^T$. Thus the PUMA robot is commanded to translate its end-effector approximately 37in and rotate its end-effector through 150°.

The proposed adaptive regulation strategy (9) is utilized to achieve the specified position control. To provide a basis for comparison, a nonadaptive controller and a classical PD control law are also used to provide this motion control. The nonadaptive regulation scheme is obtained by removing the adaptive term $\mathbf{f}$ from (9)

$$\mathbf{F} = k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \tag{12}$$
$$\dot{\mathbf{w}} = -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}}$$

while the PD controller is obtained from (12) by assuming that velocity information is available:

$$\mathbf{F} = k_1{}^{2}\gamma^2\dot{\mathbf{e}}^{76} + k_2\gamma^2\mathbf{e} \tag{13}$$

In this simulation, perfect (noise-free) velocity information is provided to the PD strategy (13). For all strategies, the controller parameters k_1, k_2, γ are set as follows: $k_1 = 10$, $k_2 = 20$, $\gamma = 5$. In (9), the adaptive gain $\mathbf{f}$ is set to zero initially and the adaptation parameters are set as follows: $\sigma_0 = 0.1$, $\beta = 100$, and $f_{max} = 500$. These gains were selected using the guidelines given in Section 3, and no attempt was made to "tune" the gains to obtain the best possible performance. Indeed, we found that excellent performance could be obtained with (9) by using a wide range of parameter values. For example, it was found that given any reasonable values (in the sense of the guidelines given in Section 3) for the parameters k_1, k_2, and γ, virtually any desired accuracy could be obtained simply by increasing the adaptation gain β. The results of this simulation are given in Figures 2a and 2b, with Figure 2a showing the response of the vertical position error for the end-effector and Figure 2b depicting the evolution of the orientation error for the end-effector for the three controllers (9), (12), and (13). The response for the other two end-effector position error components were quite similar and hence are not shown. Observe that accurate position regulation is achieved with the adaptive scheme (9). More specifically, the steady-state vertical position error and orientation error are less than 0.002in and 0.01°, respectively. As expected, implementation of the nonadaptive scheme (12) results in significantly degraded performance compared with the adaptive strategy (9): the steady-state vertical position and orientation errors are on the order of 3.3in and 5.3°, respectively, for the nonadaptive controller (12). This decrease in performance is most substantial for the position portion of the regulation task because gravity effects are most pronounced for positioning. The PD controller (13) performs somewhat better than the nonadaptive strategy (12) because it has access to full state information; for this controller the steady-state vertical position and orientation errors are on the order of 2.3in and 1.1°, respectively. Notice that the steady-state position error is still quite large because the simple PD controller (13) has no direct means of compensating for the effects of gravity loads.

The next simulation demonstrates the flexibility and modularity of the proposed adaptive regulation strategy (9). In this simulation, the Zebra Zero robot is initially at rest with joint-space position $\theta(0) = [-45°, 0°, -100°, 0°, 45°, 0°]^T$. The robot is commanded to move its end-effector 10in in the x-direction, 10in in the y-direction, and 10in in the z-direction, where all motions are specified relative to the base coordinate frame. Additionally, the robot is required to change its end-effector orientation from an initial orientation matrix R_{init} to a final orientation matrix R_{final}, where

$$R_{init} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad R_{final} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

and where each rotation matrix R is specified relative to the base frame of the robot [14]. The equivalent-angle-axis parameterization of this desired orientation change is $\psi_d = 90°$ and $\hat{\mathbf{k}}_d = [1, 0, 0]^T$. Thus the Zebra robot is commanded to translate its end-effector approximately 17.3in and rotate its end-effector through 90°.

The proposed adaptive regulation strategy (9) is utilized to achieve the specified position control. In this simulation, the controller parameters k_1, k_2, γ, $\mathbf{f}(0)$, σ_0, β, and f_{max} are set to the same values used in the previous simulation with the PUMA 560 robot, even though the PUMA arm is considerably larger and more massive than the Zebra. Note that this provides a means of examining the flexibility and modularity of the control scheme (9), and also demonstrates that the performance of the adaptive controller is quite insensitive to the particular choice of controller parameters.

The results of this simulation are given in Figures 3a and 3b, with Figure 3a showing the response of the (norm of the) position error for the end-effector and Figure 3b depicting the evolution of the orientation error for the end-effector. Observe that accurate position regulation is achieved with the adaptive scheme (9) despite the fact that the controller parameters are clearly not "tuned" to provide good performance. In this simulation, the steady-state position and orientation errors are less than 0.001in and 0.01°, respectively.

The last set of simulations demonstrates the capability of the control scheme (10),(11) to provide accurate task-space trajectory tracking. This portion of the simulation study consists of two parts: a *nominal performance study* in which no external disturbances are applied to the manipulator dynamics, and a *robustness study* where initial tracking errors and a payload-related disturbance are introduced. First we examine the capability of the control scheme (10),(11) to provide accurate task-space trajectory tracking under nominal conditions, in which no external disturbances are present. The PUMA robot is initially at rest with joint-space position $\theta(0) = [0, 90, 180, 0, -90, 0]^T$. The robot is commanded to smoothly cycle its end-effector through a sinusoidal trajectory with amplitude 20in in the x-direction, 20in in the y-direction, and 20in in the z-direction for 6 seconds according to the following temporal trajectories: $x_d(t) = x_i + 10(1 - \cos(\pi/2)t), y_d(t) = y_i + 10(1 - \cos(\pi/2)t), z_d(t) = z_i + 10(1 - \cos(\pi/2)t)$, for $t \in [0, 6]$, where x_i, y_i, z_i are the initial position coordinates of the end-effector and all motions are specified relative to the base coordinate frame (see Figure 1). Additionally, the robot is required to smoothly cycle its end-effector orientation from an initial orientation matrix R_{init} to a final orientation matrix R_{final}, where

$$R_{init} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad R_{final} = \begin{bmatrix} \sqrt{3}/2 & -1/2 & 0 \\ 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and where each rotation matrix R is specified relative to the base frame of the robot. The desired temporal trajectory for this orientation change is specified as follows: $\psi_d = 75°(1 - \cos(\pi/2)t)$ and $\hat{\mathbf{k}}_d = [0,0,1]^T$ for $t \in [0,6]$ (see Figure 1). Thus the robot is commanded to rapidly translate and rotate its end-effector through a sinusoidal trajectory with a translational amplitude of approximately 35in and a rotational amplitude of 150° for 6 seconds. The control strategy (10),(11) is utilized to achieve the specified trajectory tracking. The adaptive gains $\mathbf{f}, A, B$ are set to zero initially, while the remaining controller parameters are set as follows: $k_1 = 10$, $k_2 = 20$, $\gamma = 5$, and $\alpha_i = 0.01$, $\beta_i = 100$ for $i = 1,2,3$. Note that all of the controller parameters in the tracking strategy (10),(11) which also appear in the regulation scheme (9) are set to the values used in the regulation experiment summarized above, illustrating the modularity and flexibility inherent in this approach to manipulator motion control.

The results of this simulation are given in Figures 4a-4c, and indicate that accurate task-space trajectory tracking is achieved. For example, the maximum translational and rotational tracking errors are less than 0.05in and 0.7°, respectively, for this simulation. To illustrate the effects of adaptation, the same trajectory was tracked with the nonadaptive controller obtained by setting the adaptation gains β_i to zero in (10),(11). The performance of this nonadaptive scheme is also shown in Figures 4a-4c, and a comparison of the plots makes clear the improvement in tracking obtainable through the use of adaptation.

The next simulation in this study investigates the robustness of the controller (10),(11) to external disturbances. More specifically, the effects of an abrupt change in payload mass and of initial positioning errors are examined. To study the effect of these disturbances, the controller (10),(11) is utilized to repeat the task specified in the previous simulation, but here a 5.0in position error is introduced in each translational DOF, and the payload is abruptly reduced from 50lb to 0lb at the midpoint of the trajectory. In this simulation all of the controller parameters are set to the values used in the nominal study. The results of this simulation are given in Figures 5a and 5b, and indicate that accurate task-space trajectory tracking is achieved despite the initial position errors and the abrupt change in payload mass.

5. Experimental Results

The adaptive position regulation strategy (9) and trajectory tracking scheme (10),(11) are now applied to an industrial robot in an experimental investigation. The experimental facility utilized for this study is the New Mexico State University Robotics Laboratory and consists of a IMI Zebra Zero robot arm and the associated control computer (see Figure 1). The Zebra robot possesses a conventional design with six revolute joints configured in a "waist-shoulder-elbow-wrist" arrangement (see Figure 1). All control software is written in 'C' and is hosted on an IBM-compatible 486 personal computer. A complete description of this flexible and modular experimental testbed is beyond the scope of this paper; the interested reader is referred to [23] for such a description. It is noted, however, that the open architecture environment is designed to allow convenient implementation and experimental evaluation of advanced control schemes using the Zebra robot, and therefore provides an ideal testbed for the present investigation. Throughout the study the unit of time is second and the unit of angle is degree.

In this set of experiments, the Zebra manipulator is controlled in joint-space. In the first experiment, the Zebra manipulator is initially at rest with joint-space position $\theta(0) = [-45°, 0°, -100°, 0°, 45°, 0°]^T$, and is commanded to move to the desired final configuration $\theta_d = [45°, 90°, -100°, 0°, 45°, 0°]^T$. Thus the waist and shoulder joints are each requested to move through 90° and to initiate and terminate the motion with zero velocity, while the remaining joints are commanded to remain in their initial positions. This position control task is performed using the adaptive regulation scheme (9) and, to provide a basis for comparison, the task is also performed using the IMI controller which is provided with the Zebra manipulator. To illustrate the effect that payload variation has on controller performance, the desired regulation task is repeated for two payloads: one of 0lb and one of 2lb (recall that the Zebra manipulator is a small laboratory robot, so that a 2lb payload is a significant load). The adaptive control strategy (9) is implemented with the adaptive gain $\mathbf{f}$ initialized to zero and the remaining controller parameters set as follows: $k_1 = 10$, $k_2 = 20$, $\gamma = 5$, $\sigma_0 = 0.1$, $\beta = 100$, $f_{max} = 500$. These gains were selected using the guidelines given in Section 3, and no attempt was made to "tune" the gains to obtain the best possible performance; indeed, we found that a wide range of parameter values provided similar performance. The adaptive controller is applied to the robot with a sampling period of seven milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of seven milliseconds. It is mentioned that this relatively slow sampling rate is the fastest one available in the present experimental setup, and does not represent a limitation resulting from the computational requirements of the adaptive controller; indeed, the adaptive scheme (9) can be run at a much faster rate using the computing equipment presently in place. Note that the IMI controller operates with a sampling period of one millisecond, and therefore possesses a considerable advantage in this experiment.

The results of this set of experiments are given in Figures 6a and 6b (0lb payload) and Figures 7a and 7b (2lb payload). It can be seen from Figures 6a and 6b that the adaptive scheme (9) and IMI controller provide comparable performance if there is no payload present, despite the fact that the IMI control loop is seven times faster than the adaptive control loop. The steady-state error for both the waist joint (Figure 6a) and shoulder joint (Figure 6b) are approximately 0.1° for the adaptive scheme and 0.2° for the IMI controller. An examination of Figures 7a and 7b reveals that the adaptive controller provides virtually identical performance despite the presence of the 2lb payload, while the accuracy of the IMI controller degrades noticeably. As expected, the effect of the payload is most evident with the shoulder joint, since it is this joint which must move against gravity torques (see Figure 1b). More specifically, in the presence of the payload the shoulder joint steady-state error for the adaptive scheme (9) increases slightly to approximately 0.2°, while with the IMI controller the steady-state error for this joint increases significantly to approximately 3.0°. This experiment provides an illustration of one of the advantages of the adaptive approach to manipulator control: uniform performance in the presence of variation in payload.

In the next experiment, the adaptive trajectory tracking scheme (10),(11) is applied to the problem of tracking a user-specified trajectory for the manipulator joint angles. The Zebra manipulator is initially at rest with the same joint-space position $\theta(0)$ as in the regulation experiments. The waist joint is commanded to smoothly move from its initial position of $\theta_1 = -45°$ to an intermediate position of $\theta_1 = 45°$, and then to return to its initial position; the desired temporal trajectory for this motion is specified to be $\theta_{1d} = -45° + 45°(1 - \cos(\pi/2)t)$ for $t \in [0, 4]$. At the same time, the shoulder joint is commanded to move from $\theta_2 = 0°$ to $\theta_2 = 60°$ and then back to $\theta_2 = 0°$ according to the trajectory $\theta_{2d} = 30°(1 - \cos(\pi/2)t)$ for $t \in [0, 4]$. All of the other manipulator joints are commanded to remain in their initial positions. Thus the manipulator is required to track a large and rapid joint-space trajectory with no *a priori* information concerning the system model. The control strategy (10),(11) is utilized to achieve the specified trajectory tracking. The adaptive gains $\mathbf{f}, A, B$ are set to zero initially, while the remaining controller parameters are set as follows: $k_1 = 10$, $k_2 = 20$, $\gamma = 5$, and $\sigma_i = 0$, $\beta_i = 100$ for $i = 1, 2, 3$. Note that all of the controller parameters in the tracking strategy (10),(11) which also appear in the regulation scheme (9) are set to the values used in the regulation experiment summarized above, illustrating the modularity inherent in this approach to manipulator motion control. The adaptive controller is again applied to the robot with a sampling period of seven milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of seven milliseconds.

The results of this experiment are given in Figures 8a and 8b. During the arm motion, the joint encoder counts are recorded directly from the arm, converted into degrees, and stored in a data file. The maximum tracking errors incurred at the waist and shoulder joints are less than 0.8° and 0.7°, respectively. Figures 8a and 8b show the desired and actual trajectories of the waist and shoulder joints, respectively, and confirm that accurate joint-space trajectory tracking is achieved.

6. Conclusions

This paper has introduced a new approach to adaptive motion control of uncertain robot manipulators. The primary contributions of the paper are as follows:

1. The position regulation problem has been studied, and an adaptive solution to this problem has been proposed which does not require joint velocity measurements or knowledge of the gravity model for the manipulator or payload. To the best of our knowledge, this is the first control scheme capable of compensating for both of these sources of uncertainty (other than the preliminary result presented by the authors in [11]). The proposed controller is simple, easy to implement, ensures uniform boundedness of all signals in the presence of bounded external disturbances, and possess desirable convergence properties: asymptotic convergence if no disturbances are present and convergence to an arbitrarily small neighborhood of zero in the presence of disturbances. It is noted that even the state feedback version of this result (i.e., the controller obtained from (9) by replacing $\mathbf{w}$ with $\dot{\mathbf{e}}$ in the case that velocity is accurately measurable) represents a new contribution.
2. It has been demonstrated that the regulation scheme can be modified to provide accurate tracking control through the introduction of additional feedforward elements in the control law. The resulting tracking strategy is very general and computationally efficient, requires virtually no information regarding the manipulator dynamic model, is implementable without velocity measurements, and ensures uniform boundedness of all signals and (ultimately) arbitrarily accurate tracking.
3. The results of extensive computer simulations and hardware experiments have been presented to demonstrate the performance achievable with the proposed controllers. The simulations show that the adaptive schemes consistently and significantly outperform their nonadaptive counterparts; additionally, these simulations indicate that the proposed regulation scheme provides performance superior to a standard PD controller,

despite the fact that we supply the PD controller with perfect (no noise) velocity information. The hardware experiments confirm that the proposed approach provides a simple and effective means of obtaining high performance motion control. For example, these experiments show that the adaptive regulation scheme is more accurate than a well-tuned industrial controller.

Future work will focus on the application of the proposed approach of controller development to the problem of robot compliant motion control, and will initially involve an attempt to generalize the results obtained in [20] so that velocity measurements are no longer required. Progress along these lines is reported in [24].

7. Acknowledgments

The research described in this paper was supported in part through contracts with the Army Research Office, Sandia National Laboratories, the Department of Energy (WERC), and the Jet Propulsion Laboratory. The authors would also like to acknowledge the helpful suggestions of an anonymous reviewer which improved the paper considerably.

8. References

1. Takegaki, M. and S. Arimoto, "A New Feedback Method for Dynamic Control of Manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, 1981, pp. 119-125

2. Klafterm, R., T. Chmielewski, and M. Negin, *Robotic Engineering: An Integrated Approach*, Prentice Hall, Englewood Cliffs, NJ, 1989

3. Tomei, P., "Adaptive PD Controller for Robot Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 565-570

4. Wen, J., K. Kreutz-Delgado, and D. Bayard, "Lyapunov Function-Based Control Laws for Revolute Robot Arms: Tracking Control, Robustness, and Adaptive Control", *IEEE Transactions on Automatic Control*, Vol. 37, No. 2, 1992, pp. 231-237

5. Kelly, R., "Comments on "Adaptive PD Controller for Robot Manipulators' ", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 1, 1993, pp. 117-119

6. Nicosia, S. and P. Tomei, "Robot Control by Using Only Joint Position Measurements", *IEEE Transactions on Automatic Control*, Vol. 35, No. 9, 1990, pp. 1058-1061

7. Yuan, J. and Y. Stepanenko, "Robust Control of Robotic Manipulators Without Velocity Feedback", *International Journal of Robust and Nonlinear Control*, Vol. 1, 1991, pp. 203-213

8. Canudas de Wit, C., N. Fixot, and K. Astrom, "Trajectory Tracking in Robot Manipulators via Nonlinear Estimated State Feedback", *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 1, 1992, pp. 138-144

9. Berghuis, H. and H. Nijmeijer, "Global Regulation of Robots Using Only Position

Measurements" *Systems and Control Letters*, Vol. 21, 1993, pp. 289-293

10. Feng, W. and I. Postlethwaite, "A Simple Robust Control Scheme for Robot Manipulators With Only Joint Position Measurements", *International Journal of Robotics Research*, Vol. 12, No. 5, 1993, pp. 490-496

11. Colbaugh, R., K. Glass, and E. Barany, "Adaptive Output Stabilization of Manipulators", *Proc. 33rd IEEE Conference on Decision and Control*, Orlando, FL, December 1994

12. Seraji, H., "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 4, 1989, pp. 472-490

13. Glass, K., R. Colbaugh, D. Lim, and H. Seraji, "Real-time Collision Avoidance for Redundant Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 3, 1995

14. Spong, M. and M. Vidyasagar, *Robot Dynamics and Control*, Wiley, New York, 1989

15. Berghuis, H., R. Ortega, and H. Nijmeijer, "A Robust Adaptive Robot Controller", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, 1993, pp. 825-830

16. Koditschek, D., "Adaptive Strategies for the Control of Natural Motion", *Proc. 24th IEEE Conference on Decision and Control*, Ft. Lauderdale, FL, December 1985

17. Colbaugh, R., "Adaptive Point-to-Point Motion Control of Manipulators", *International Journal of Robotics and Automation*, Vol. 9, No. 2, 1994, pp. 51-61

18. Khatib, O., "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation", *IEEE Journal of Robotics and Automation*, Vol. 3, No. 1, 1987, pp. 43-53

19. Colbaugh, R. and K. Glass, "Cartesian Control of Redundant Robots", *Journal of Robotic Systems*, Vol. 6, No. 4, 1989, pp. 427-459

20. Colbaugh, R., H. Seraji, and K. Glass, "Adaptive Compliant Motion Control for Dexterous Manipulators", *International Journal of Robotics Research*, Vol. 14, No. 3, 1995

21. Vidyasagar, M., *Nonlinear Systems Analysis*, Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1993

22. Ioannou, P., "Robust Adaptive Controller with Zero Residual Tracking Errors", *IEEE Transactions on Automatic Control*, Vol. 31, No. 8, 1986, pp. 773-776

23. Glass, K. and R. Colbaugh, "A Robust Open-Architecture Computer Simulation and Experimental Testbed for Manipulator Controller Development", Robotics Laboratory Report 94-01, New Mexico State University, February 1994

24. Colbaugh, R. and K. Glass, "Adaptive Impedance Control of Manipulators Without Velocity Measurements", Robotics Laboratory Report 95-04, New Mexico State University, March 1995

25. Colbaugh, R. and K. Glass, "Robust Adaptive Regulation of Manipulators: Theory and Experiments", Robotics Laboratory Report 94-10, New Mexico State University, December 1994

26. Corless, M. and G. Leitmann, "Continuous State Feedback Guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems", *IEEE Transactions on Au-*

tomatic Control, Vol. 26, No. 5, 1981, pp. 1139-1144

27. Colbaugh, R. and K. Glass, "Adaptive Tracking Control of Manipulators Using Only Position Measurements", Robotics Laboratory Report 95-02, New Mexico State University, January 1995

9. Appendix A: Proof of Theorem 1

In this Appendix, the proof of Theorem 1 is given; additional details concerning this analysis can be found in the report [25]. We consider first the case in which no external disturbances are acting. Applying the control law (9) to the *nominal* manipulator dynamics (3) (with $\mathbf{d} = \mathbf{0}$) yields the closed-loop error dynamics

$$\begin{aligned} &H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi} + \mathbf{G}(\mathbf{x}_d) - \mathbf{G}(\mathbf{x}) = \mathbf{0} \\ &\dot{\mathbf{w}} = -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \\ &\dot{\mathbf{f}} = -\sigma\mathbf{f} + \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}) \end{aligned} \qquad (A1)$$

where $\mathbf{\Phi} = \mathbf{f} - \mathbf{G}(\mathbf{x}_d)$. Let $U(\mathbf{x})$ denote the gravitational potential energy of the manipulator. Consider the Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2\beta}\mathbf{\Phi}^T\mathbf{\Phi} + U(\mathbf{x}) - U(\mathbf{x}_d) + \mathbf{G}^T(\mathbf{x}_d)\mathbf{e} \qquad (A2)$$

and note that V_2 is a positive-definite and radially-unbounded function of $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$ and $\mathbf{\Phi}$ if γ is chosen sufficiently large (for example, if γ is chosen large enough then the term $k_2\gamma^2\mathbf{e}^T\mathbf{e}/2 + U(\mathbf{x}) - U(\mathbf{x}_d) + \mathbf{G}^T(\mathbf{x}_d)\mathbf{e}$ can be shown to be positive-definite in $\mathbf{e}$ using an analysis analogous to the one given in [3]).

Computing the derivative of (A2) along (A1) and simplifying as in the proof of Lemma 1 yields

$$\begin{aligned} \dot{V}_2 \le & -(\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\ & + 2\lambda_{max}(H) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 \\ & + \frac{1}{\beta}\mathbf{\Phi}^T(\dot{\mathbf{f}} - \beta[\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}]) + \parallel \mathbf{G}(\mathbf{x}_d) - \mathbf{G}(\mathbf{x}) \parallel\parallel \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w} \parallel \\ \le & -(\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - (\frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma}) \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\ & + 2\lambda_{max}(H) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{M}{\gamma} \parallel \mathbf{w} \parallel\parallel \mathbf{e} \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 \end{aligned} \qquad (A3)$$

where M is a positive scalar constant satisfying $M \parallel \mathbf{x}_d - \mathbf{x} \parallel \ge \parallel \mathbf{G}(\mathbf{x}_d) - \mathbf{G}(\mathbf{x}) \parallel \forall \mathbf{x}_d, \mathbf{x}$ (the boundedness of the partial derivatives of $\mathbf{G}$ ensures that such an M exists), and where it is assumed that f_{max} is chosen so that $\| \mathbf{G}(\mathbf{x}_d) \| \leq f_{max} \quad \forall \mathbf{x}_d$. Define $\mathbf{z}$ as in Lemma 1 and Q_2 as

$$Q_2 = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma} & 0 & -\frac{M}{2\gamma} \\ 0 & (\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) & -\lambda_{max}(H) \\ -\frac{M}{2\gamma} & -\lambda_{max}(H) & k_1\gamma \end{bmatrix}$$

and note that Q_2 is positive-definite if γ is chosen large enough. Then the following bound on $\dot{V}_2$ in (A3) can be obtained through routine manipulation:

$$\dot{V}_2 \leq -(\lambda_{min}(Q_2) - \frac{\alpha_2}{\gamma} {V_2}^{1/2}) \| \mathbf{z} \|^2 \qquad (A4)$$

where α_2 is a scalar constant which does not increase as γ is increased. Let $c_2(t) = \lambda_{min}(Q_2) - \alpha_2 {V_2}^{1/2}(t)/\gamma$ and choose γ large enough so that $c_2(0) > 0$ (this is always possible). Then (A4) implies that $c_2(0) \leq c_2(t) \quad \forall t$, so that $\dot{V}_2 \leq -c_2(0) \| \mathbf{z} \|^2$. Standard arguments can then be used to show that all signals are bounded and that $\mathbf{z}$ (and therefore $\mathbf{e}, \dot{\mathbf{e}}, \mathbf{w}$) converges to zero asymptotically [e.g., 21].

We consider next the case in which external disturbances are present. Applying the adaptive scheme (9) to the *disturbed* manipulator dynamics (3) (with $\mathbf{d}$ nonzero) yields the closed-loop error dynamics

$$\begin{aligned} H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi}_d = \mathbf{0} \\ \dot{\mathbf{w}} = -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \\ \dot{\mathbf{f}} = -\sigma\mathbf{f} + \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}) \end{aligned} \qquad (A5)$$

where $\mathbf{\Phi}_d = \mathbf{f} - \mathbf{G} - \mathbf{d}$. Define the Lyapunov function candidate

$$V_3 = V_1 + \frac{1}{2\beta}\mathbf{\Phi}_d^T\mathbf{\Phi}_d \qquad (A6)$$

and note that V_3 is a positive-definite and radially-unbounded function of $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$ and $\mathbf{\Phi}_d$ if γ is chosen sufficiently large. Differentiating (A6) along (A5) and simplifying as in the proof of Lemma 1 yields

$$\begin{aligned} \dot{V}_3 \leq -(\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) \| \dot{\mathbf{e}} \|^2 - \frac{k_2^2\gamma}{k_1} \| \mathbf{e} \|^2 - k_1\gamma \| \mathbf{w} \|^2 \\ + 2\lambda_{max}(H) \| \mathbf{w} \| \| \dot{\mathbf{e}} \| + \frac{k_2 k_{cc}}{k_1\gamma} \| \mathbf{e} \| \| \dot{\mathbf{e}} \|^2 + \frac{k_{cc}}{\gamma} \| \mathbf{w} \| \| \dot{\mathbf{e}} \|^2 \\ + \frac{1}{\beta}\mathbf{\Phi}_d^T(\dot{\mathbf{f}} - \dot{\mathbf{G}} - \dot{\mathbf{d}} - \beta(\dot{\mathbf{e}}2+86\frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w})) \end{aligned} \qquad (A7)$$

Defining $\mathbf{z}$ as in Lemma 1 and Q_3 as

$$Q_3 = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} & 0 & 0 \\ 0 & (\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_{min}(H) & -\lambda_{max}(H) \\ 0 & -\lambda_{max}(H) & k_1\gamma \end{bmatrix}$$

(and noticing that Q_3 is positive-definite if γ is chosen large enough) permits the following bound on $\dot{V}_3$ in (A7) to be obtained:

$$\begin{aligned} \dot{V}_3 \leq -\mathbf{z}^T Q_3 \mathbf{z} + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel \parallel \mathbf{e} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel \parallel \dot{\mathbf{e}} \parallel^2 \\ - \frac{\gamma}{2}\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{1}{\beta}\mathbf{\Phi}_d^T(\sigma\mathbf{f} + \dot{\mathbf{G}} + \dot{\mathbf{d}}) \end{aligned} \qquad (A8)$$

If f_{max} is selected so that $\parallel \mathbf{G}(\mathbf{x}) + \mathbf{d}(t) \parallel \leq f_{max} \;\; \forall \mathbf{x}, t$ then the following bound on $\dot{V}_3$ in (A8) can be derived through routine manipulation:

$$\dot{V}_3 \leq -(\lambda_{min}(Q_3) - \frac{\alpha_3}{\gamma}{V_3}^{1/2}) \parallel \mathbf{z} \parallel^2 - \frac{\sigma}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2 + \frac{\eta}{\beta}$$

where α_3 and η are scalar constants which do not increase as γ, β increase.

The remainder of the proof will utilize the ultimate boundedness result presented by the authors in [11,20], which is based on the work of Corless and Leitmann [26]. Toward this end, it is convenient to bound V_3 in (A6) in the following manner:

$$\lambda_1 \parallel \mathbf{z} \parallel^2 + \frac{1}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2 \leq V_3 \leq \lambda_3 \parallel \mathbf{z} \parallel^2 + \frac{1}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2$$

where the λ_i are positive scalar constants independent of β. Now choose two scalar constants V_M, V_m so that $V_M > V_m \geq V_3(0)$, and define $c_M = \lambda_{min}(Q_3) - \alpha_3 {V_M}^{1/2}/\gamma$; then choose γ large enough so that $c_M > 0$ (this is always possible). Let $\delta^* = \lambda_3\eta/c_M + 2\max(2f_{max}^2, \eta/\sigma_0)$ and choose β_0 so that $\delta^*/\beta_0 < V_m$ (this is always possible). Then selecting $\beta \geq \beta_0$ ensures that if $V_m \leq V_3 \leq V_M$ then $\dot{V}_3 < 0$. This condition together with $V_M > V_m \geq V_3(0)$ implies that $V_3(t) \leq V_M \;\forall t$, so that $c_3(t) = \lambda_{min}(Q_3) - \alpha_3 {V_3}^{1/2}(t)/\gamma > c_M > 0 \;\; \forall t$ and

$$\dot{V}_3 \leq -c_M \parallel \mathbf{z} \parallel^2 - \frac{\sigma}{2(\beta_0 + \Delta\beta)} \parallel \mathbf{\Phi}_d \parallel^2 + \frac{\eta}{(\beta_0 + \Delta\beta)}$$

where $\Delta\beta = \beta - \beta_0$ and it is assumed that β is chosen so that $\Delta\beta > 0$. It follows immediately that $\parallel \mathbf{z} \parallel$ and $\parallel \mathbf{\Phi}_d \parallel$ are uniformly bounded. Moreover, the ultimate boundedness result given in [11,20] is now directly applicable and permits the conclusion that $\| \mathbf{z} \|$ converges asymptotically to a closed ball of radius $r^2 = \delta^*/(\lambda_1(\beta_0 + \Delta\beta))$. This completes the proof of Theorem 1.

10. Appendix B: Proof of Theorem 2

In this Appendix, the proof of Theorem 2 is given; additional details concerning this analysis can be found in the report [27]. Applying the control law (10) to the *disturbed* manipulator dynamics (3) (with $\mathbf{d}$ nonzero) yields the closed-loop system dynamics

$$\begin{aligned} &H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi}_f + \Phi_A\ddot{\mathbf{x}}_d + \Phi_B\dot{\mathbf{x}}_d + V_{cd}\dot{\mathbf{e}} = \mathbf{0} \\ &\dot{\mathbf{w}} = -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \end{aligned} \qquad (B1)$$

where $\mathbf{\Phi}_f = \mathbf{f} - \mathbf{G} - \mathbf{d}$, $\Phi_A = A - H$, $\Phi_B = B - V_{cd}$, and the notation $V_{cd} = V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d)$ is introduced. Note that in obtaining (B1) the Coriolis/centripetal term was expanded as follows: $V_{cc}\dot{\mathbf{x}}_d = V_{cd}\dot{\mathbf{x}} = V_{cd}\dot{\mathbf{x}}_d - V_{cd}\dot{\mathbf{e}}$. Consider the Lyapunov function candidate

$$V_4 = V_1 + \frac{1}{2\beta_1}\mathbf{\Phi}_f^T\mathbf{\Phi}_f + \frac{1}{2}\mathrm{tr}[\frac{1}{\beta_2}\Phi_A\Phi_A^T + \frac{1}{\beta_3}\Phi_B\Phi_B^T] \qquad (B2)$$

and note that V_4 is a positive-definite and radially-unbounded function of $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$, $\mathbf{\Phi}_f$, Φ_A, and Φ_B if γ is chosen sufficiently large. Computing the derivative of (B2) along (B1) and simplifying using the adaptation laws (11) yields

$$\begin{aligned} \dot{V}_4 = &-(\gamma - \frac{k_2}{k_1\gamma})\dot{\mathbf{e}}^T H\dot{\mathbf{e}} - \frac{k_2^2\gamma}{k_1} \| \mathbf{e} \|^2 - k_1\gamma \| \mathbf{w} \|^2 + 2\mathbf{w}^T H\dot{\mathbf{e}} \\ &+ \frac{1}{\gamma}\dot{\mathbf{e}}^T[\frac{k_2}{k_1}V_{cc}\mathbf{e} - V_{cc}\mathbf{w}] - \mathbf{q}^T V_{cd}\dot{\mathbf{e}} + \frac{1}{\beta_1}\mathbf{\Phi}_f^T(\dot{\mathbf{\Phi}}_f - \beta_1\mathbf{q}) \\ &+ \mathrm{tr}[\frac{1}{\beta_2}\Phi_A(\dot{\Phi}_A - \beta_2\mathbf{q}\ddot{\mathbf{x}}_d^T)^T + \frac{1}{\beta_3}\Phi_B(\dot{\Phi}_B - \beta_3\mathbf{q}\dot{\mathbf{x}}_d^T)^T] \\ \le &-((\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_{min}(H) - k_{cd}) \| \dot{\mathbf{e}} \|^2 - \frac{k_2^2\gamma}{k_1} \| \mathbf{e} \|^2 - k_1\gamma \| \mathbf{w} \|^2 \\ &+ (2\lambda_{max}(H) + \frac{k_{cd}}{\gamma} + \frac{k_{cc}v_{max}}{\gamma}) \| \mathbf{w} \| \| \dot{\mathbf{e}} \| + \frac{k_2}{k_1\gamma}(k_{cc}v_{max} + k_{cd}) \| \mathbf{e} \| \| \dot{\mathbf{e}} \| \\ &+ \frac{k_2 k_{cc}}{k_1\gamma} \| \mathbf{e} \| \| \dot{\mathbf{e}} \|^2 + \frac{k_{cc}}{\gamma} \| \mathbf{w} \| \| \dot{\mathbf{e}} \|^2 - \frac{\sigma_1}{2\beta_1} \| \mathbf{\Phi}_f \|^2 - \frac{\sigma_2}{2\beta_2} \| \Phi_A \|_F^2 - \frac{\sigma_3}{2\beta_3} \| \Phi_B \|_F^2 \\ &- \frac{\gamma}{6}\lambda_{min}(H) \| \dot{\mathbf{e}} \|^2 - \frac{\sigma_1}{2\beta_1} \| \mathbf{\Phi}_f \|^2 + \frac{\eta_1}{\beta_1} \| \dot{\mathbf{e}} \| \| \mathbf{\Phi}_f \| + \frac{\eta_2}{\beta_1} \| \mathbf{\Phi}_f \| \\ &- \frac{\gamma}{6}\lambda_{min}(H) \| \dot{\mathbf{e}} \|^2 - \frac{\sigma_2}{2\beta_2} \| \Phi_A \|_F^2 + \frac{\eta_3}{\beta_2} \| \dot{\mathbf{e}} \| \| \Phi_A \|_F + \frac{\eta_4}{\beta_2} \| \Phi_A \|_F \\ &- \frac{\gamma}{6}\lambda_{min}(H) \| \dot{\mathbf{e}} \|^2 - \frac{\sigma_3}{2\beta_3} \| \Phi_B \|_F^2 + 2\frac{\eta_5}{\beta_3}88\| \dot{\mathbf{e}} \| \| \Phi_B \|_F + \frac{\eta_6}{\beta_3} \| \Phi_B \|_F \end{aligned} \qquad (B3)$$

where k_{cd} is an upper bound on V_{cd} (i.e., $\| V_{cd} \|_F \leq k_{cd} \ \ \forall \mathbf{x}$), v_{max} is an upper bound on $\| \dot{\mathbf{x}}_d \|$, and the η_i are positive scalar constants.

Observe that the following inequalities can be obtained through routine manipulation:

$$-\frac{\gamma}{6}\lambda_{min}(H) \| \dot{\mathrm{e}} \|^2 - \frac{\sigma_1}{2\beta_1} \| \mathbf{\Phi}_f \|^2 + \frac{\eta_1}{\beta_1} \| \dot{\mathrm{e}} \| \| \mathbf{\Phi}_f \| + \frac{\eta_2}{\beta_1} \| \mathbf{\Phi}_f \| \leq \frac{\eta_7}{\beta_1}$$
$$-\frac{\gamma}{6}\lambda_{min}(H) \| \dot{\mathrm{e}} \|^2 - \frac{\sigma_2}{2\beta_2} \| \Phi_A \|_F^2 + \frac{\eta_3}{\beta_2} \| \dot{\mathrm{e}} \| \| \Phi_A \|_F + \frac{\eta_4}{\beta_2} \| \Phi_A \|_F \leq \frac{\eta_8}{\beta_2}$$
$$-\frac{\gamma}{6}\lambda_{min}(H) \| \dot{\mathrm{e}} \|^2 - \frac{\sigma_3}{2\beta_3} \| \Phi_B \|_F^2 + \frac{\eta_5}{\beta_3} \| \dot{\mathrm{e}} \| \| \Phi_B \|_F + \frac{\eta_6}{\beta_3} \| \Phi_B \|_F \leq \frac{\eta_9}{\beta_3}$$

where η_7, η_8, η_9 are positive scalar constants which do not increase as the β_i are increased. These inequalities permit the following upper bound on $\dot{V}_4$ in (B3) to be established:

$$\begin{aligned}\dot{V}_4 \leq & -((\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_{min}(H) - k_{cd}) \| \dot{\mathrm{e}} \|^2 - \frac{k_2^2\gamma}{k_1} \| \mathrm{e} \|^2 - k_1\gamma \| \mathbf{w} \|^2 \\ & + (2\lambda_{max}(H) + \frac{k_{cd}}{\gamma} + \frac{k_{cc}v_{max}}{\gamma}) \| \mathbf{w} \| \| \dot{\mathrm{e}} \| + \frac{k_2}{k_1\gamma}(k_{cc}v_{max} + k_{cd}) \| \mathrm{e} \| \| \dot{\mathrm{e}} \| \\ & + \frac{k_2 k_{cc}}{k_1\gamma} \| \mathrm{e} \| \| \dot{\mathrm{e}} \|^2 + \frac{k_{cc}}{\gamma} \| \mathbf{w} \| \| \dot{\mathrm{e}} \|^2 - \frac{\sigma_1}{2\beta_1} \| \mathbf{\Phi}_f \|^2 \\ & - \frac{\sigma_2}{2\beta_2} \| \Phi_A \|_F^2 - \frac{\sigma_3}{2\beta_3} \| \Phi_B \|_F^2 + \frac{\eta_7}{\beta_1} + \frac{\eta_8}{\beta_2} + \frac{\eta_9}{\beta_3} \end{aligned} \qquad (B4)$$

Let $\mathbf{z} = [\| \mathrm{e} \| \ \ \| \dot{\mathrm{e}} \| \ \ \| \mathbf{w} \|]^T$, $\Psi = [\| \mathbf{\Phi}_f \| \ \ \| \Phi_A \|_F \ \ \| \Phi_B \|_F]^T$, $\beta_{min} = \min(\beta_i)$, $\beta_{max} = \max(\beta_i)$, and

$$Q = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} & -\frac{k_2}{2k_1\gamma}(k_{cc}v_{max} + k_{cd}) & 0 \\ -\frac{k_2}{2k_1\gamma}(k_{cc}v_{max} + k_{cd}) & (\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_{min}(H) - k_{cd} & -\lambda_{max}(H) - \frac{k_{cd}}{2\gamma} - \frac{k_{cc}v_{max}}{2\gamma} \\ 0 & -\lambda_{max}(H) - \frac{k_{cd}}{2\gamma} - \frac{k_{cc}v_{max}}{2\gamma} & k_1\gamma \end{bmatrix}$$

and note that Q is positive-definite if γ is chosen large enough. If β_{max}/β_{min} is fixed then there exist positive scalar constants η_{10}, η_{11} that do not increase as γ and β_{min} increase, and positive scalar constants λ_i independent of γ and β_{min}, such that V_4 and $\dot{V}_4$ in (B2) and (B4) can be bounded as

$$\lambda_1 \| \mathbf{z} \|^2 + \frac{\lambda_2}{\beta_{min}} \| \Psi \|^2 \leq V_4 \leq \lambda_3 \| \mathbf{z} \|^2 + \frac{\lambda_4}{\beta_{min}} \| \Psi \|^2$$
$$\dot{V}_4 \leq -(\lambda_{min}(Q) - \frac{\eta_{10}}{\gamma} {V_4}^{1/2}) \| \mathbf{z} \|^2 - \frac{\lambda_5}{\beta_{min}} \| \Psi \|^2 + \frac{\eta_{11}}{\beta_{min}}$$

Now choose two scalar constants V_M, V_m so that $V_M > V_m \geq V_4(0)$, and define $c_M = \lambda_{min}(Q) - \eta_{10}{V_M}^{1/2}/\gamma$; then choose γ large enough so that $c_M > 0$ (this is always possible).

Let $\delta = \max(\lambda_3/c_M, \lambda_4/\lambda_5)$ and choose β_0 so that $\eta_{11}\delta/\beta_0 < V_m$ (this is always possible). Then selecting $\beta_{min} \geq \beta_0$ ensures that if $V_m \leq V_4 \leq V_M$ then $\dot{V}_4 < 0$. This condition together with $V_M > V_m \geq V_4(0)$ implies that $V_4(t) \leq V_M\ \forall t$, so that $c(t) = \lambda_{min}(Q) - \eta_{10}{V_4}^{1/2}(t)/\gamma > c_M > 0\ \ \forall t$ and

$$\dot{V}_4 \leq -c_M \parallel \mathbf{z} \parallel^2 - \frac{\lambda_5}{(\beta_0 + \Delta\beta)} \parallel \mathbf{\Psi} \parallel^2 + \frac{\eta_{11}}{(\beta_0 + \Delta\beta)}$$

where $\Delta\beta = \beta_{min} - \beta_0$ and it is assumed that β_{min} is chosen so that $\Delta\beta > 0$. An analysis identical to the one used in the proof of Theorem 1 now applies and permits the conclusion that $\parallel \mathbf{z} \parallel, \parallel \mathbf{\Psi} \parallel$ are uniformly bounded, which implies that $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$, $\mathbf{f}$, A, and B are uniformly bounded. Moreover, $\parallel \mathbf{z} \parallel, \parallel \mathbf{\Psi} \parallel$ converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where

$$r_1 = \left(\frac{\delta\eta_{11}}{\lambda_1(\beta_0 + \Delta\beta)} \right)^{1/2}$$

$$r_2 = \left(\frac{\delta\eta_{11}}{\lambda_2} \right)^{1/2}$$

Observe that the radius of the ball to which $\parallel \mathbf{z} \parallel^2 = \parallel \mathbf{e} \parallel^2 + \parallel \dot{\mathbf{e}} \parallel^2 + \parallel \mathbf{w} \parallel^2$ is guaranteed to converge can be decreased as desired simply by increasing $\Delta\beta$. This completes the proof of Theorem 2.

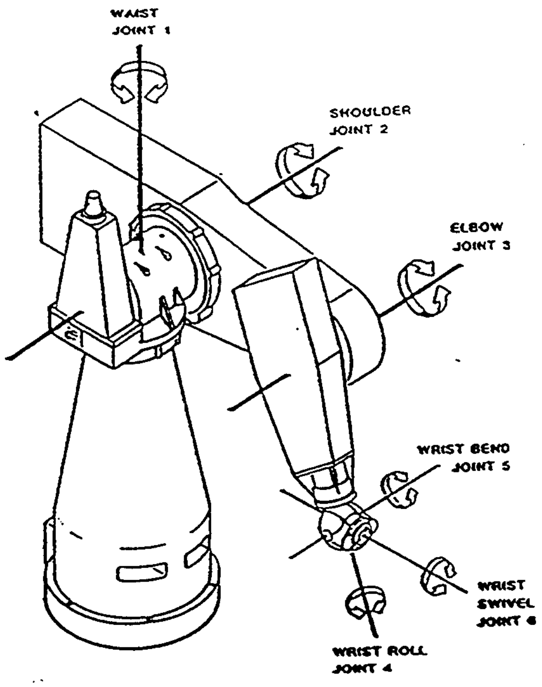


Figure 1a: Illustration of PUMA manipulator

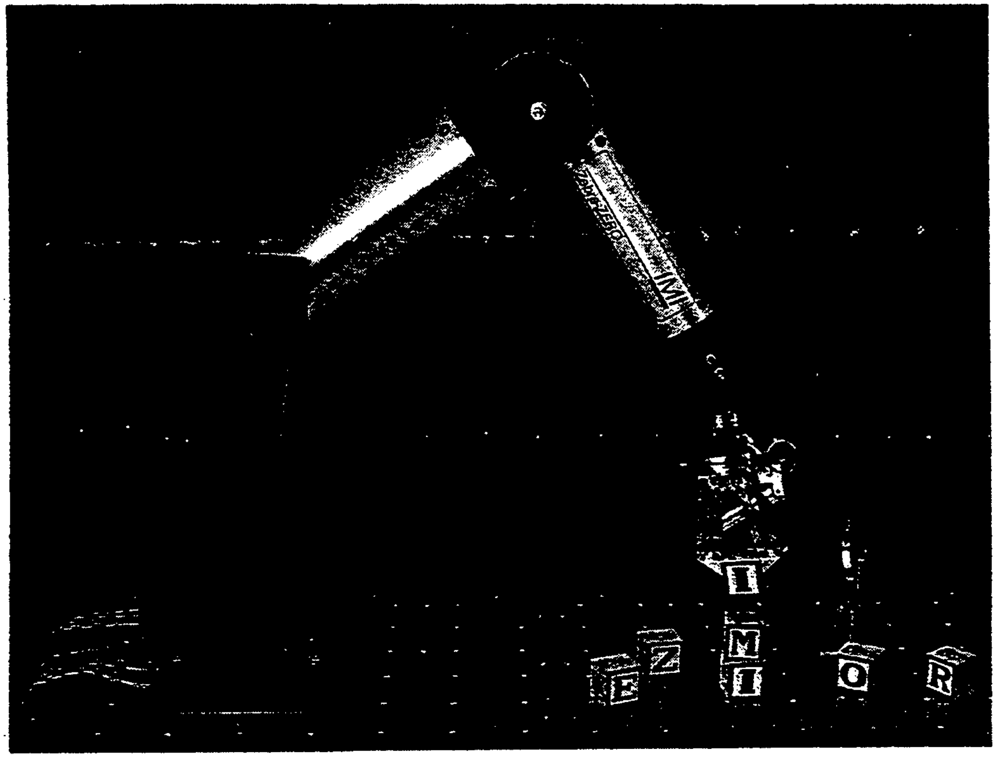


Figure 1b: Illustration of Zebra Zero manipulator

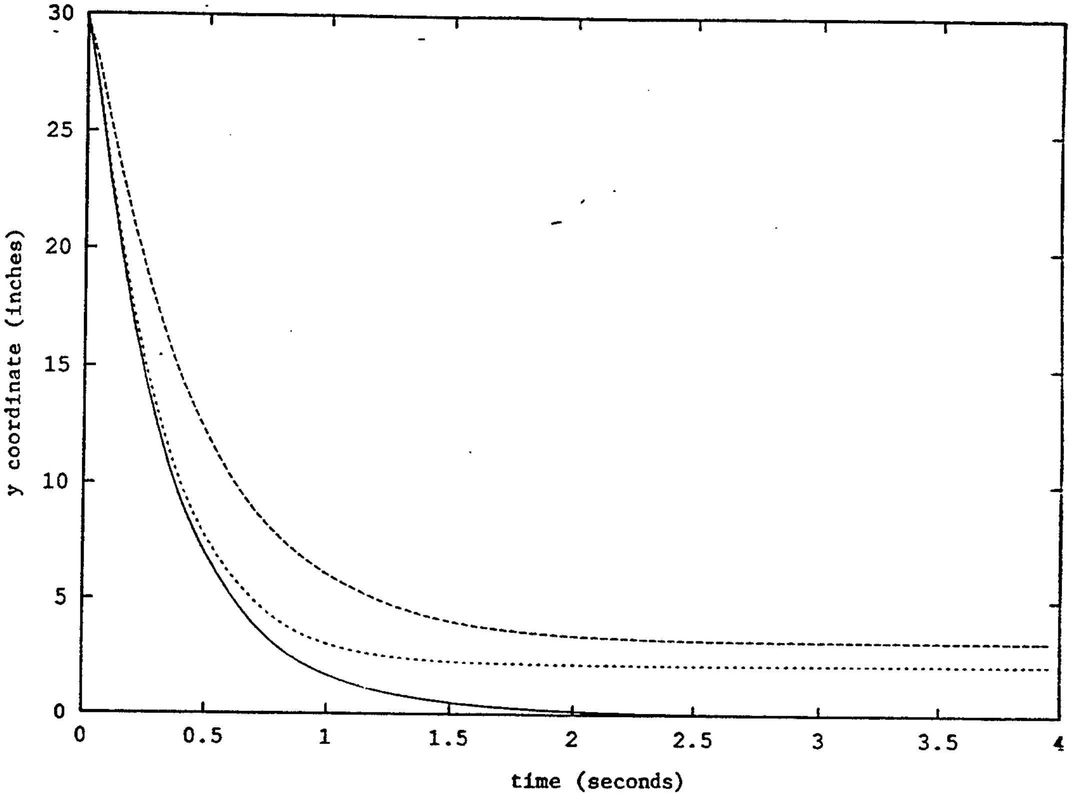


Figure 2a: Adaptive (solid), nonadaptive (dashed), and PD (dotted) controlled response of end-effector coordinate y in PUMA arm simulation study (position regulation)

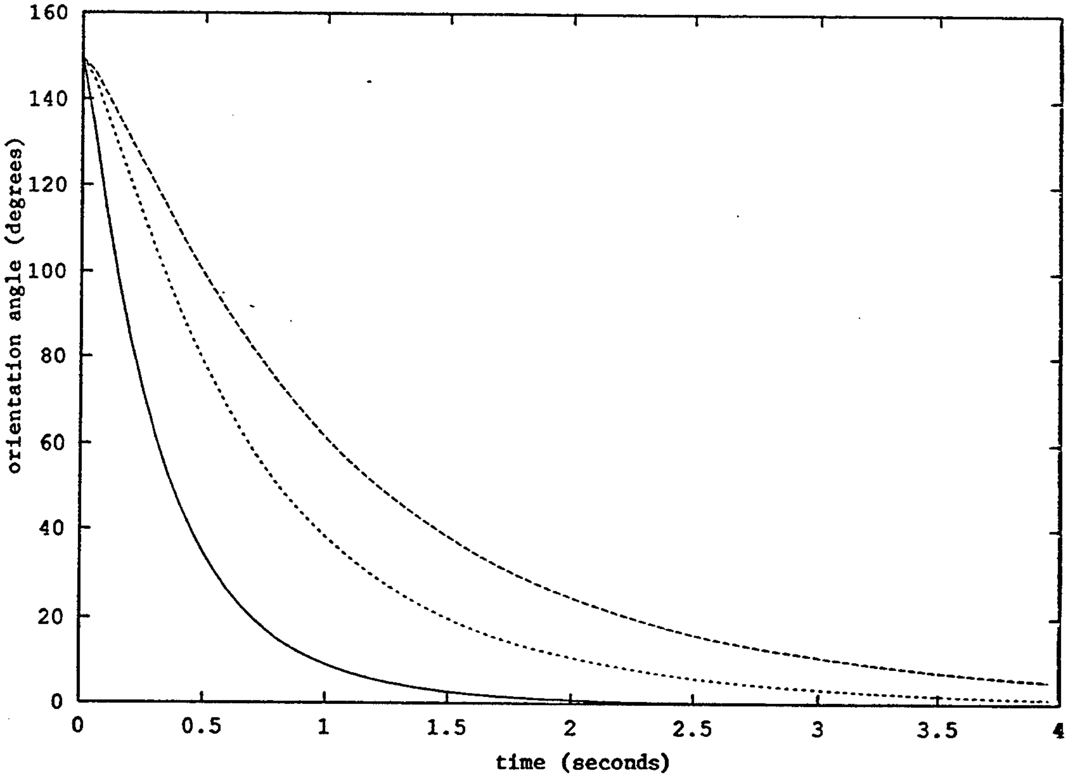


Figure 2b: Adaptive (solid), nonadaptive (dashed), and PD (dotted) controlled response of end-effector orientation angle ψ in PUMA arm simulation study (position regulation)

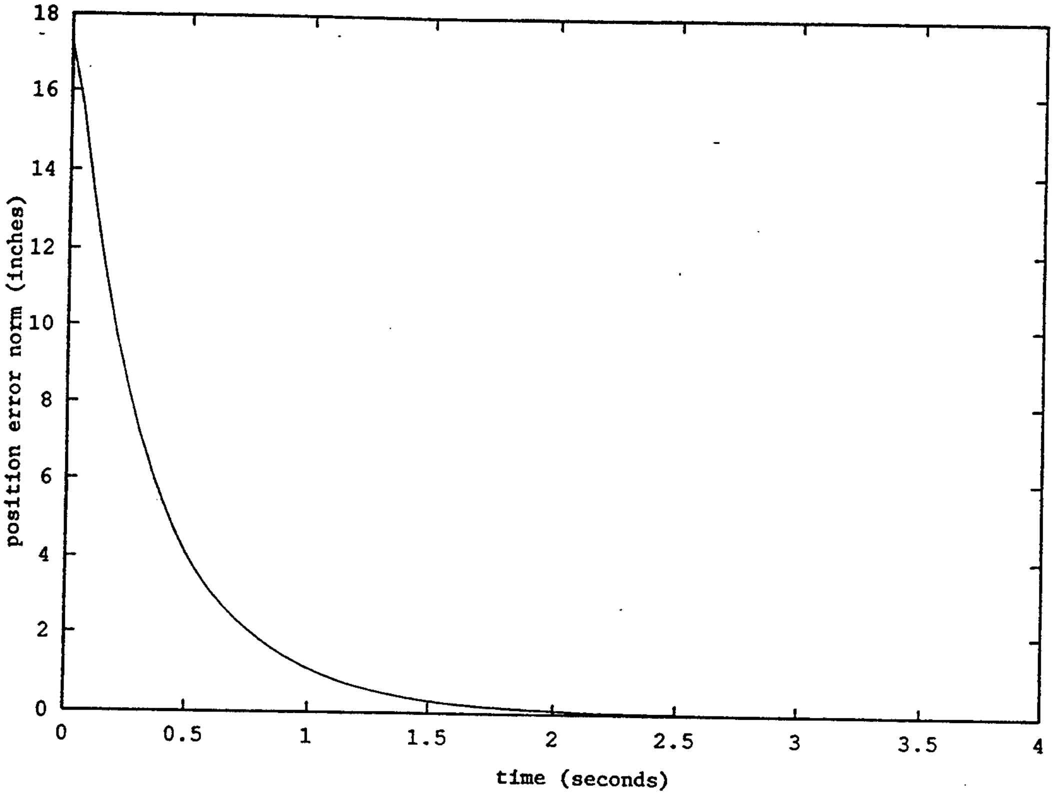


Figure 3a: Response of Zebra robot end-effector position error norm in simulation study (position regulation)

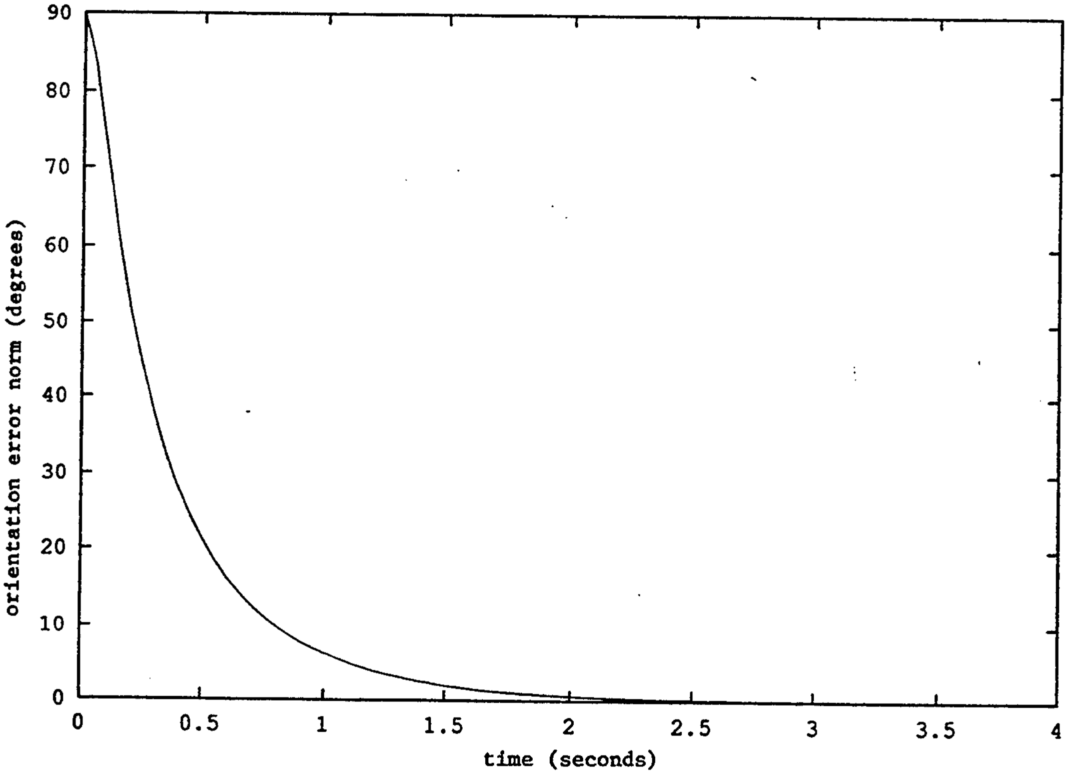


Figure 3b: Response of Zebra robot end-effector orientation error norm in simulation study (position regulation)

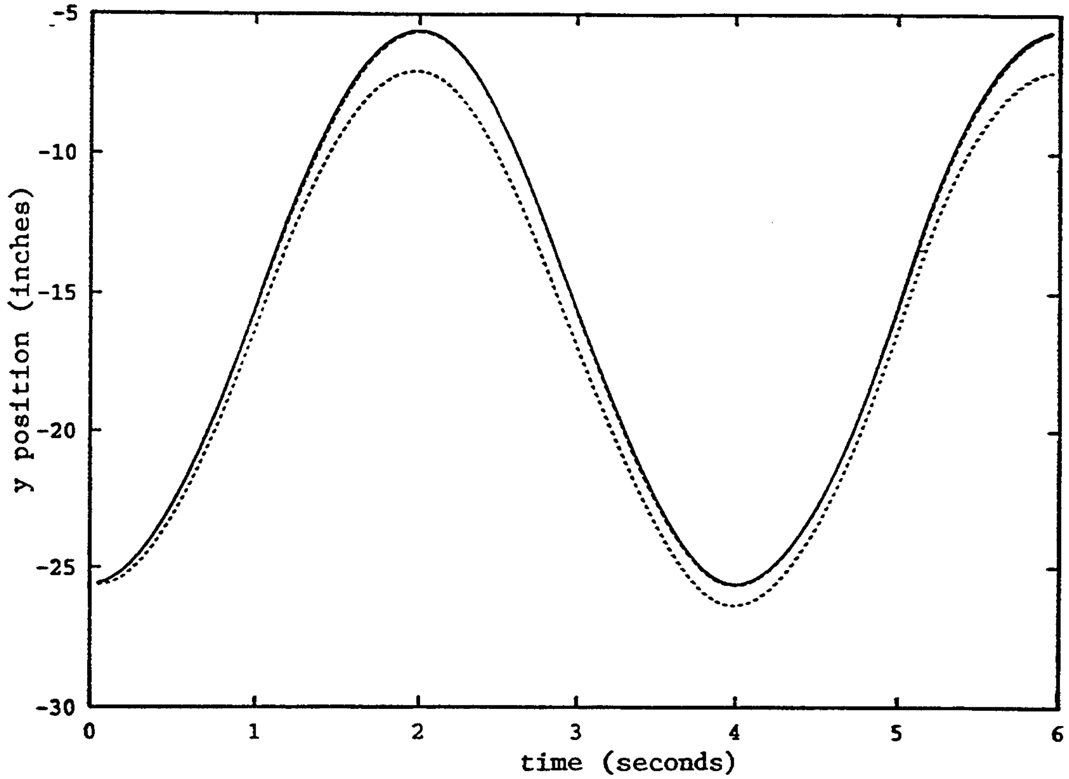


Figure 4a: Desired (solid), adaptive (dashed), and nonadaptive (dotted) response of end-effector coordinate y in PUMA arm simulation study (trajectory tracking, nominal case)

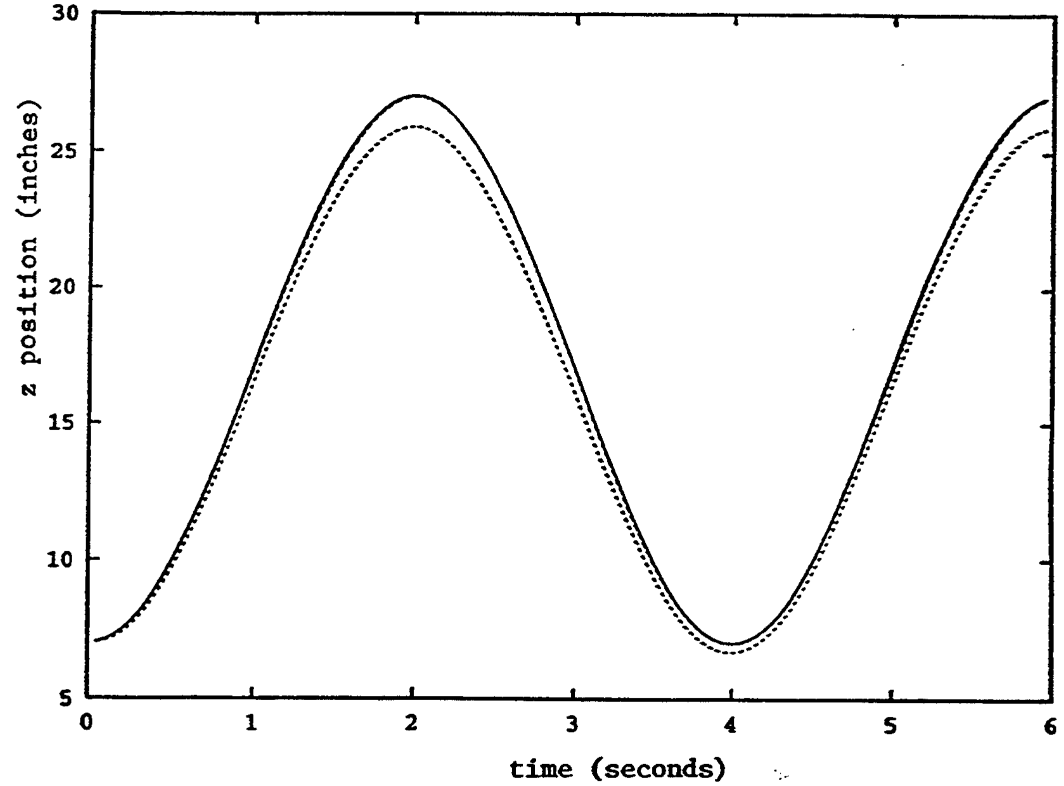


Figure 4b: Desired (solid), adaptive (dashed), and nonadaptive (dotted) response of end-effector coordinate z in PUMA arm simulation study (trajectory tracking, nominal case)

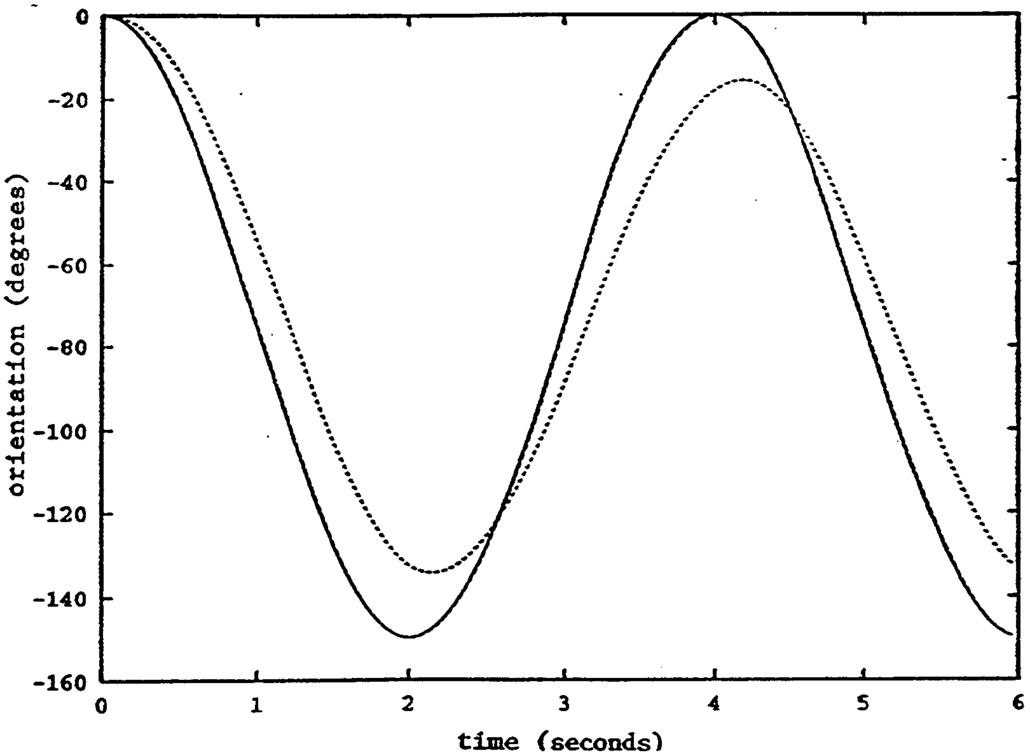


Figure 4c: Desired (solid), adaptive (dashed), and nonadaptive (dotted) response of end-effector orientation angle ψ in PUMA arm simulation study (trajectory tracking, nominal case)

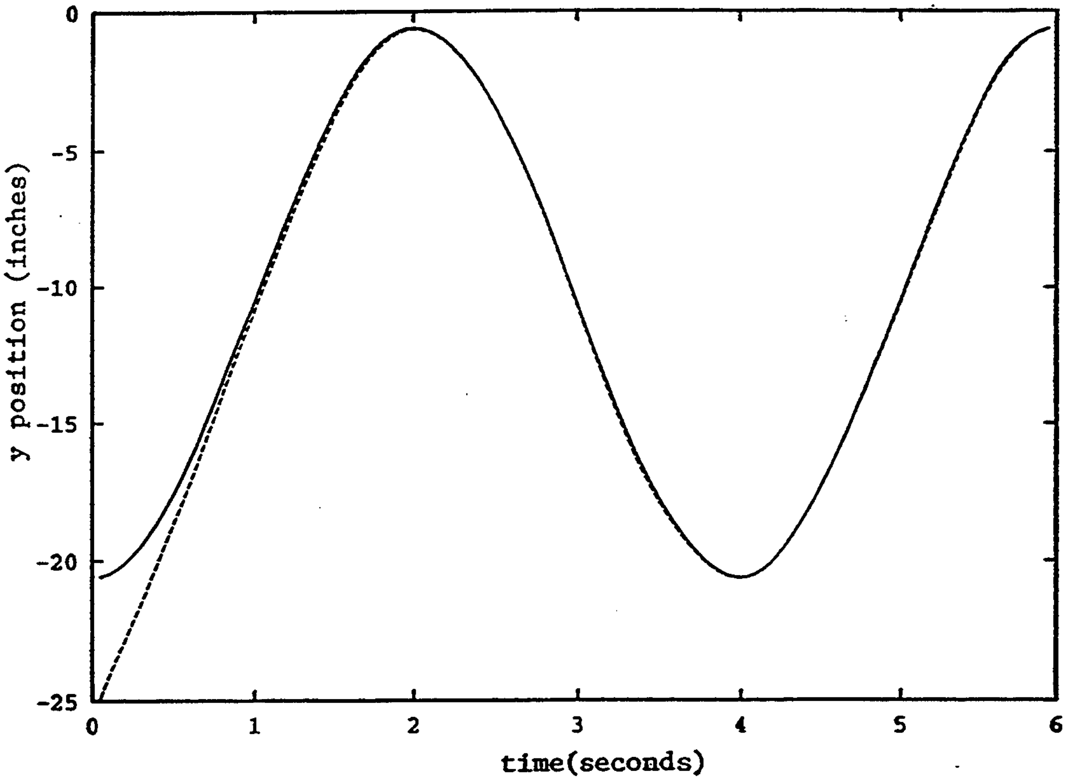


Figure 5a: Desired (solid) and adaptive (dashed) response of end-effector coordinate y in PUMA arm simulation study (trajectory tracking, robust case)

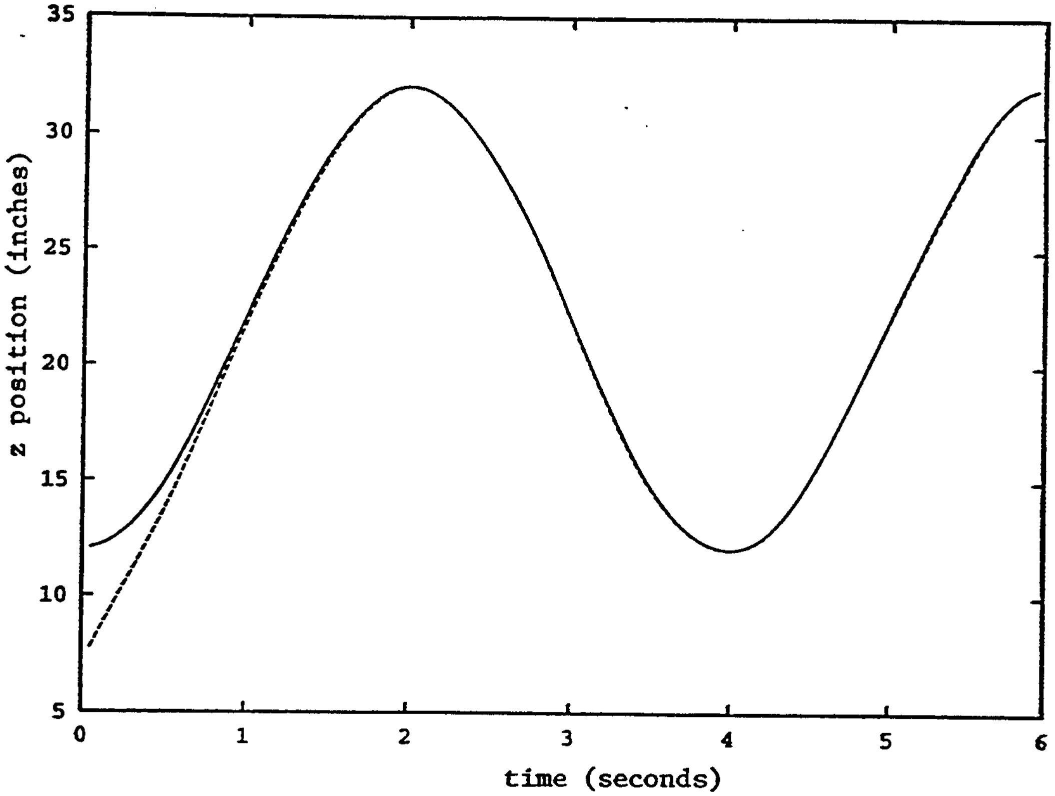


Figure 5b: Desired (solid) and adaptive (dashed) response of end-effector coordinate z in PUMA arm simulation study (trajectory tracking, robust case)

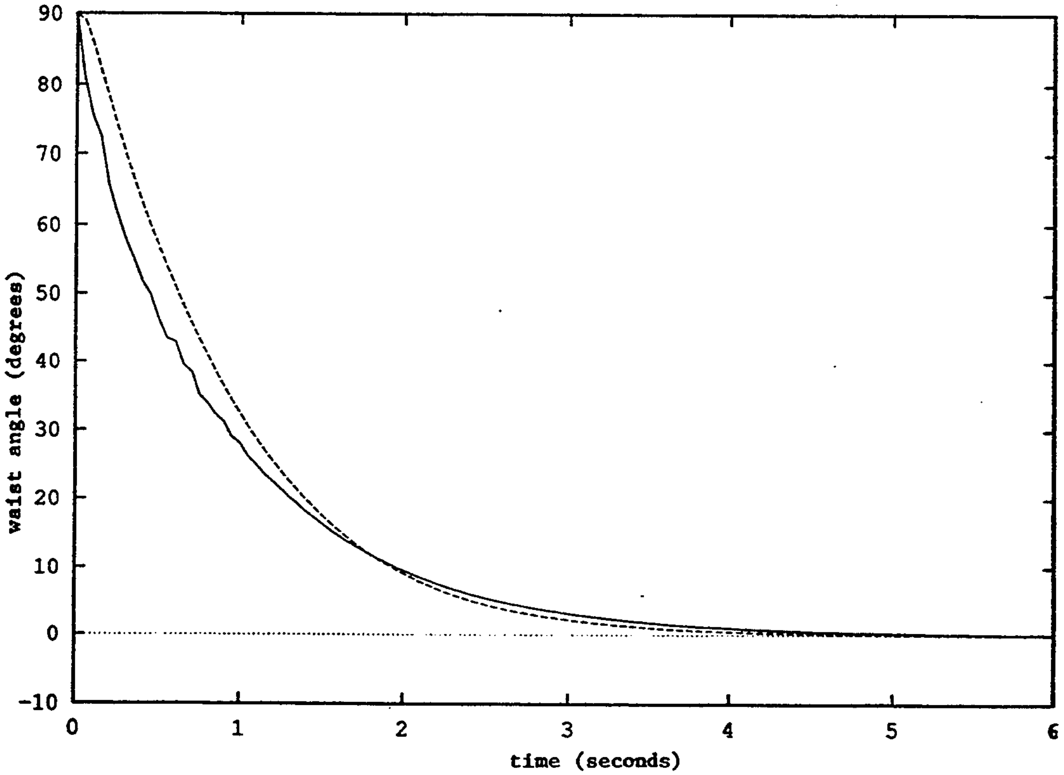


Figure 6a: Adaptive (solid) and IMI (dashed) controlled response of Zebra waist angle in experimental study (position regulation, no payload)

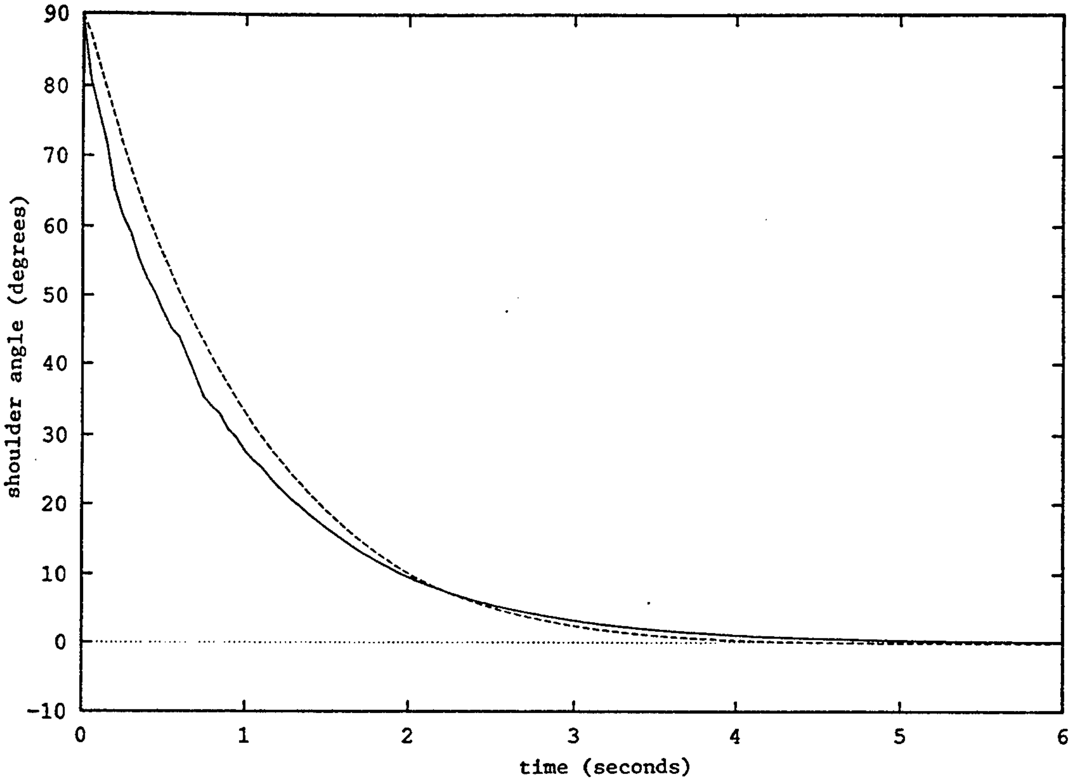


Figure 6b: Adaptive (solid) and IMI (dashed) controlled response of Zebra shoulder angle in experimental study (position regulation, no payload)

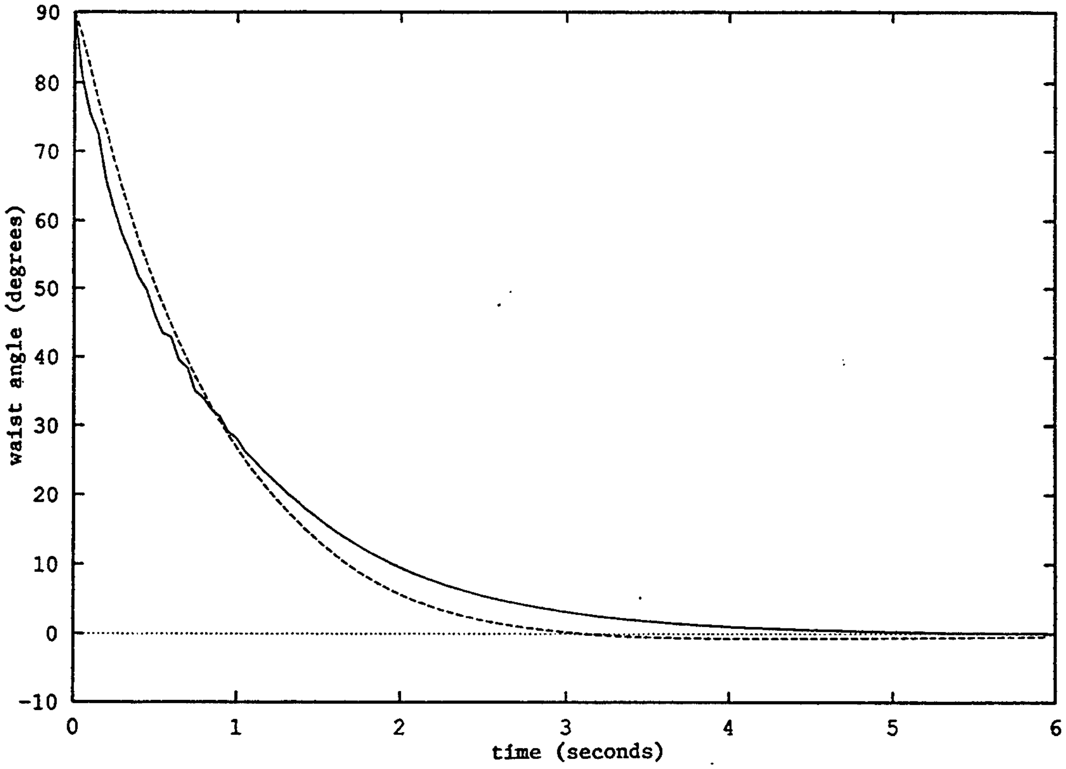


Figure 7a: Adaptive (solid) and IMI (dashed) controlled response of Zebra waist angle in experimental study (position regulation, payload)

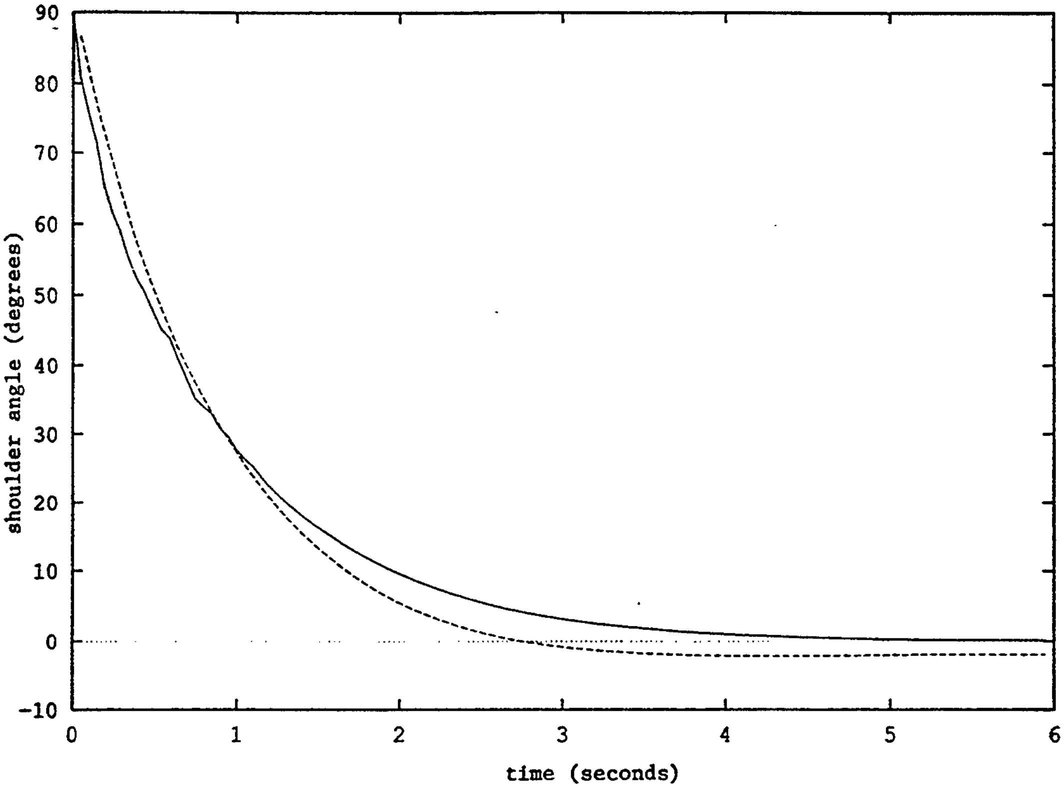


Figure 7b: Adaptive (solid) and IMI (dashed) controlled response of Zebra shoulder angle in experimental study (position regulation, payload)

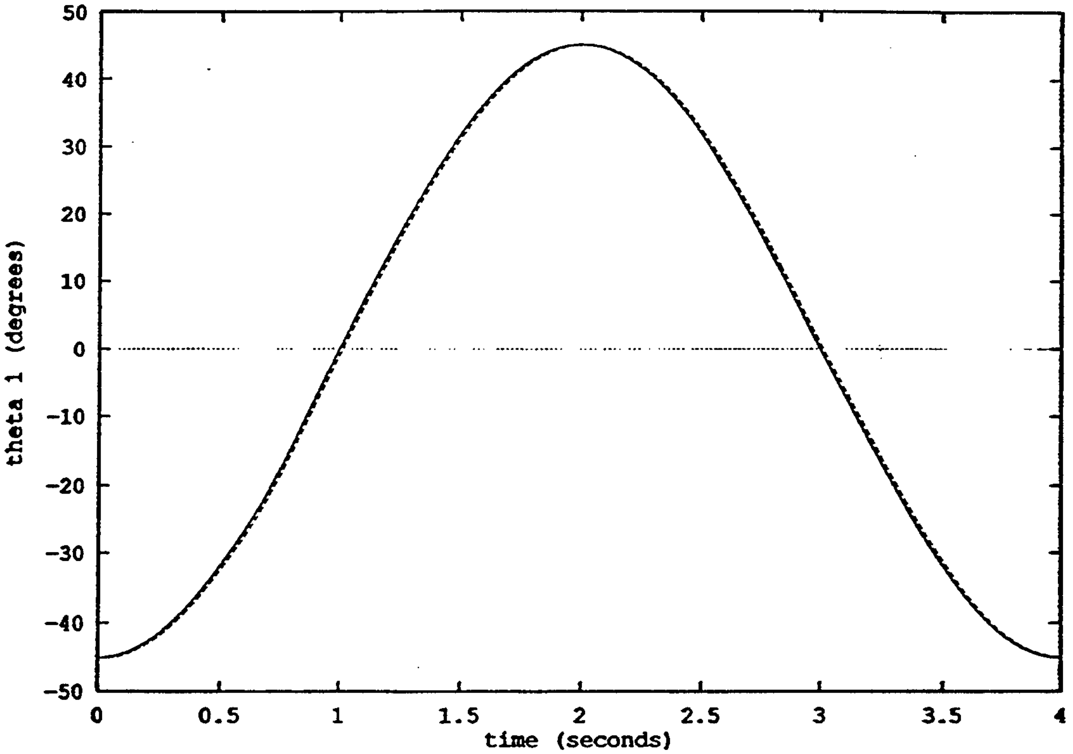


Figure 8a: Desired (solid) and actual (dashed) response of Zebra waist angle in experimental study (trajectory tracking)

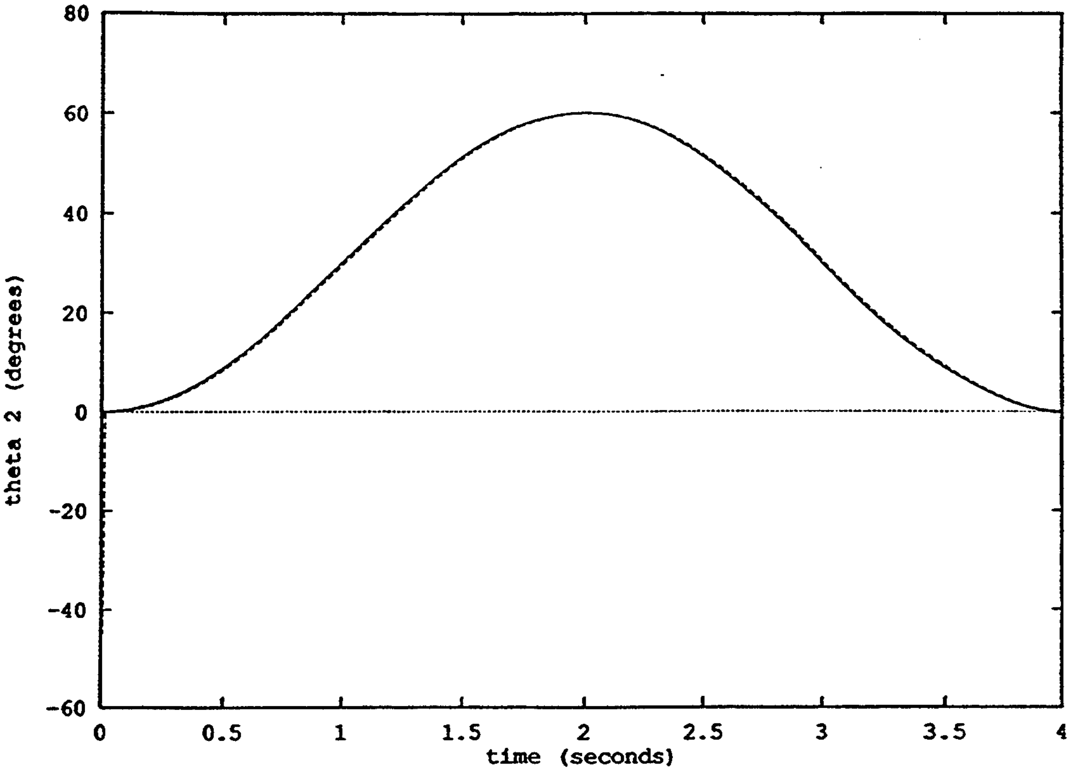


Figure 8b: Desired (solid) and actual (dashed) response of Zebra shoulder angle in experimental study (trajectory tracking)

Communication

Adaptive Compliant Motion Control for Dexterous Manipulators

Richard Colbaugh
Department of Mechanical Engineering
New Mexico State University
Las Cruces, New Mexico 88003

Homayoun Seraji
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

Kristin Glass
Department of Mechanical Engineering
New Mexico State University
Las Cruces, New Mexico 88003

Abstract

This article presents two adaptive schemes for compliant motion control of dexterous manipulators. The first scheme is developed using an adaptive impedance control approach for torque-controlled manipulators, whereas the second strategy is an adaptive admittance controller for position-controlled manipulators. The proposed controllers are very general and computationally efficient, as they do not require knowledge of the manipulator dynamic model or parameter values of the manipulator or the environment and are implemented without calculation of the inverse dynamics or inverse kinematic transformation. It is shown that the control strategies are globally stable in the presence of bounded disturbances and that in the absence of disturbances the ultimate bound on the size of the system errors can be made arbitrarily small. The capabilities of the proposed control schemes are illustrated through both computer simulations and laboratory experiments with a dexterous Robotics Research Corporation seven-degrees-of-freedom (DOF) manipulator.

1. Introduction

Manipulators are subject to interaction forces whenever they perform tasks involving motion that is constrained by the environment, such as assembling parts or executing grinding or finishing operations. In such motions the interaction forces must be accommodated rather than resisted, so as to comply with the environmental constraints. Recognizing the importance of compliant motion for robots, many researchers have investigated this problem in recent years. This intensive study has produced two basic approaches to achieving compliant motion: force control(Raibert and Craig 1981) and impedance control (Hogan 1985). The first approach is based on the observation that when the manipulator end-effector is in contact with the environment, the Cartesian space of the end-effector coordinates can be naturally decomposed into a "position subspace" and a "force subspace"; these two subspaces correspond to the Cartesian directions in which the end-effector is, respectively, free to move and constrained by the environment. The force control approach to compliant motion is then to define and track a position (and orientation) trajectory in the position subspace and a force (and moment) trajectory in the force subspace. Alternatively, the impedance control approach proposes that the control objective should be the regulation of the mechanical impedance of the robot end-effector. Thus, the goal of the impedance controller is to maintain a desired dynamic relationship between the end-effector position and the end-effector/environment contact force. It is noted that this distinction between force control and impedance control can be viewed as primarily a philosophical one and that in implementation the two approaches are closely related (Volpe and Khosla 1993).

During the past decade there has been considerable interest in developing adaptive control strategies for robot manipulators. This attention has resulted in the design of several high-performance unconstrained motion controllers, and the success of these schemes has stimulated interest in applying adaptive control theory to the problem of robot compliant motion control (e.g., Niemeyer and Slotine 1991; Lu and Meng 1991; Carelli and Kelly 1991; Lozano and Brogliato 1992; Colbaugh et al. 1993b; Jean and Fu 1993). Virtually all of the adaptive compliant motion controllers proposed to date have been developed by assuming that the structure of the manipulator dynamics is known a priori and that only the

The International Journal of Robotics Research,
Vol. 14, No. 3, June 1995, pp. 270–280,

constant inertial parameters, which appear linearly in the model, are unknown. We will call these schemes *model-based adaptive controllers* because of their reliance on information concerning the manipulator dynamic model. Although the model-based approach has yielded adaptive schemes for which global stability can be demonstrated, there are potential difficulties associated with this method of controller development. For instance, the design and implementation of these controllers requires precise knowledge of the structure of the entire robot dynamic model, which is often unrealistic in practice, and the resulting control law can be computationally intensive. Recent studies have indicated that the transient behavior of unconstrained motion control schemes designed using the model-based adaptive approach can be undesirable and that they can lack robustness to unmodeled dynamics, sensor noise, and other disturbances (e.g., Reed and Ioannou 1989); it is expected that these problems may be compounded in the case of constrained motion because of the interaction with the environment.

It should also be mentioned that in the design of the model-based adaptive compliant motion control strategies cited earlier, it is assumed that the manipulator joint actuators can be accessed directly so that joint torque is the control input. Although this control problem is certainly important and is addressed in this article as well, it is useful to recognize that typically the manipulator is equipped with a position control system. In such cases it is of interest to consider the problem of specifying position set points for this "inner-loop" controller so that the desired compliant behavior is realized. This approach has the advantage of being readily implementable with industrial robots and as a consequence is often adopted in practice. However, there has been relatively little work in which the stability of the complete system is demonstrated or in which adaptation is introduced in the "outer loop" to provide improved performance.

This article presents two new adaptive compliant motion control schemes designed to overcome the difficulties associated with previous strategies. The control laws are derived directly in task space and do not require knowledge of the manipulator dynamic model or parameter values for the manipulator or the environment. As a consequence, the schemes are very general and computationally efficient and are implementable with both nonredundant and redundant manipulators. The first scheme is an impedance controller that adaptively generates the joint torques required to provide the desired dynamic relationship between end-effector position and contact force. The second compliant motion control strategy is an adaptive admittance controller for position-controlled manipulators. In this force control scheme, position set points for the inner-loop position controller are adaptively specified to ensure that the desired end-effector/environment contact force is achieved. It is shown that both control strategies are globally stable in the presence of bounded disturbances and that in the absence of disturbances the ultimate bound on the size of the controller errors can be made arbitrarily small. The effectiveness of the proposed schemes is illustrated through both computer simulations and laboratory experiments with a Robotics Research Corporation Model K-1207 redundant manipulator.

2. Preliminaries

Let $\mathbf{y}$ define the position and orientation of the robot end effector relative to a fixed user-defined reference frame, and note that, in the most general case, the elements of $\mathbf{y}$ are local coordinates for some smooth manifold $\mathcal{M}$ of dimension m. The forward kinematic and differential kinematic maps between the robot joint coordinates $\theta \in \mathcal{N}$ and the end-effector coordinates $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{h}(\theta), \qquad \dot{\mathbf{y}} = J(\theta)\dot{\theta} \tag{1}$$

where $\mathcal{N}$ is a smooth manifold of dimension n, $\mathbf{h} : \mathcal{N} \rightarrow \mathcal{M}$, and $J \in \Re^{m \times n}$ is the end-effector Jacobian matrix.

Because the compliant motion control objectives are stated most naturally in terms of task space variables, we will study the compliant control problem directly in task space. If the manipulator is nonredundant (so that $m = n$) and is in a region of the workspace where J has full rank, then the elements of $\mathbf{y}$ are generalized coordinates for the manipulator, and this formulation presents no difficulties. A task space formulation can also be realized if the manipulator is kinematically redundant (when $m < n$) by utilizing, for example, the configuration control approach (Seraji 1989). In what follows, we will consider nonredundant and redundant robots together and formulate the compliant motion control problem in terms of a set of n generalized coordinates $\mathbf{x}$ obtained by augmenting $\mathbf{y}$ with $n - m$ kinematic functions that define some auxiliary task to be performed by the manipulator (Seraji 1989). To retain generality, we shall require only that the elements of $\mathbf{x}$ are local coordinates for some smooth manifold $\mathcal{X}$ of dimension n and that the kinematic relationship between θ and $\mathbf{x}$ is known and continuous and can be written in a form analogous to (1):

$$\mathbf{x} = \mathbf{h}_a(\theta), \qquad \dot{\mathbf{x}} = J_a(\theta)\dot{\theta} \tag{2}$$

where $\mathbf{h}_a : \mathcal{N} \rightarrow \mathcal{X}$ and $J_a \in \Re^{n \times n}$. Observe that for $\mathbf{x}$ to be a valid generalized coordinate vector, the elements of $\mathbf{x}$ must be independent in the region of interest; thus, it will be assumed in our development that J_a is of full rank. Note also that if $\mathcal{N}$ is compact (as is the case with a manipulator possessing revolute and/or finite-travel prismatic joints) then the continuity of $\mathbf{h}_a$ implies that $\mathcal{X}$ is compact. In what follows it will be assumed that $\mathcal{N}$ is compact, and this fact will prove useful in the controller development.

Consider the manipulator dynamic model written in terms of the generalized coordinates $\mathbf{x}$:

$$\mathbf{F} = H(\mathbf{x})\ddot{\mathbf{x}} + V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}(\mathbf{x}) + \mathbf{P} + \mathbf{d}(\mathbf{x}, \dot{\mathbf{x}}, t), \qquad (3)$$

where $\mathbf{F} \in \Re^n$ is the generalized force associated with $\mathbf{x}$, $H \in \Re^{n \times n}$ is the manipulator inertia matrix, $V_{cc} \in \Re^{n \times n}$ quantifies Coriolis and centripetal acceleration effects, $\mathbf{G} \in \Re^n$ is the vector of gravity forces, and $\mathbf{P} \in \Re^n$ is the vector of forces and moments exerted by the end effector on the environment. The term $\mathbf{d} \in \Re^n$ is a vector of bounded but otherwise arbitrary disturbances that can represent unmodeled state-dependent effects or time-dependent disturbances. The properties of the system (3) that are utilized in the controller development are summarized in the following lemma.

LEMMA 1. For any set of generalized coordinates $\mathbf{x}$, the dynamic model terms $H, \mathbf{G}$ are bounded functions of $\mathbf{x}$ whose time derivatives $\dot{H}, \dot{\mathbf{G}}$ are also bounded in $\mathbf{x}$ and depend linearly on $\dot{\mathbf{x}}$; the matrix H is symmetric and positive definite; and the matrices H and V_{cc} are related by $\dot{H} = V_{cc} + V_{cc}^T$. In addition, the generalized force $\mathbf{F}$ and the physically realizable joint-space control torque $\mathbf{T} \in \Re^n$ satisfy the equation $\mathbf{T} = {J_a}^T\mathbf{F}$. Finally, $V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\mathbf{u} = V_{cc}(\mathbf{x}, \mathbf{u})\dot{\mathbf{x}}$ $\forall \mathbf{u}$, and if $\mathbf{u}$ is bounded and $\dot{\mathbf{u}}$ grows at most linearly with $\dot{\mathbf{x}}$, then $V_{cc}(\mathbf{x}, \mathbf{u})$ is bounded and $\dot{V}_{cc}(\mathbf{x}, \mathbf{u})$ grows linearly with $\dot{\mathbf{x}}$.

Proof. All of the properties except the last one are well known if joint angles are used as generalized coordinates and are proved for arbitrary generalized coordinates in Stepanenko and Yuan (1992). The last property of the Coriolis/centripetal matrix V_{cc} is easy to see by writing V_{cc} in terms of Christoffel symbols. □

Observe that in view of the relationship $\mathbf{T} = {J_a}^T\mathbf{F}$, the assumption that J_a is full rank is equivalent to the assumption that $\mathbf{F}$ is always realizable as a control input through the proper specification of $\mathbf{T}$. Therefore, our formulation of the manipulator compliant motion control problem directly in task space is justified, and the focus of the subsequent discussion is on the adaptive control of the system (3). Our approach to the design and analysis of suitable controllers is based on Lyapunov stability theory. The following lemma summarizes the properties of a certain class of Lyapunov functions and will be of direct relevance in our development.

LEMMA 2. Consider the coupled dynamic system $\dot{\mathbf{x}}_1 = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, t)$. $\dot{\mathbf{x}}_2 = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, t)$. Let $V(\mathbf{x}_1, \mathbf{x}_2, t)$ be a Lyapunov function for the system with the properties

$$\lambda_1 \parallel \mathbf{x}_1 \parallel^2 + \lambda_2 \parallel \mathbf{x}_2 \parallel^2 \leq V \leq \lambda_3 \parallel \mathbf{x}_1 \parallel^2 + \lambda_4 \parallel \mathbf{x}_2 \parallel^2$$
$$\dot{V} \leq -\lambda_5 \parallel \mathbf{x}_1 \parallel^2 - \lambda_6 \parallel \mathbf{x}_2 \parallel^2 + \epsilon,$$

where ϵ and the λ_i are positive scalar constants. Define $\delta = \max(\lambda_3/\lambda_5, \lambda_4/\lambda_6)$ and $r_i = (\delta\epsilon/\lambda_i)^{1/2}$ for $i = 1, 2$. Then for any initial state $\mathbf{x}_1(0), \mathbf{x}_2(0)$ the system will evolve so that $\mathbf{x}_1(t), \mathbf{x}_2(t)$ are uniformly bounded and converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where $B_{r_i} = \{\mathbf{x}_i \ : \parallel \mathbf{x}_i \parallel \leq r_i\}$ (see Corless [1990] for a discussion of exponential convergence to a closed ball).

Proof. Straightforward application of the global exponential convergence theorem of (Corless 1990) gives the result. □

3. Adaptive Impedance Controller

In this section the proposed adaptive impedance controller for torque-controlled manipulators is developed. When the end effector of the manipulator is in contact with the environment, the objective of impedance control is to cause the end effector to respond to contact forces according to some user-defined dynamics. For example, the desired dynamic relationship between the end-effector position $\mathbf{x}$, the end-effector reference trajectory $\mathbf{x}_r \in \mathcal{X}$, and the end-effector/environment contact force P is often specified as follows:

$$M(\ddot{\mathbf{x}}_r - \ddot{\mathbf{x}}) + C(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}) + K(\mathbf{x}_r - \mathbf{x}) = \mathrm{P}. \qquad (4)$$

where $\mathbf{x}_r$ is bounded and twice differentiable, and the matrices $M, C, K \in \Re^{n \times n}$ are specified by the user so that the dynamic system (4) possesses the desired characteristics. The proposed impedance control system consists of two subsystems: a simple filter that characterizes the desired performance (4), and an adaptive task space control scheme that generates the control input $\mathbf{F}$ to the manipulator dynamics (3) that ensures the desired performance is realized. Recall that it is assumed that J_a is full rank, and this assumption together with the relationship $\mathrm{T} = {J_a}^T\mathbf{F}$ ensures that this strategy is implementable.

3.1. Control Scheme

Observe that the impedance control objective can be realized by causing the end-effector position $\mathbf{x}$ to track closely the desired impedance trajectory $\mathbf{x}_d \in \mathcal{X}$. defined as the solution to the differential equation:

$$M\ddot{\mathbf{x}}_d + C\dot{\mathbf{x}}_d + K\mathbf{x}_d = -\mathbf{P}_m + M\ddot{\mathbf{x}}_r + C\dot{\mathbf{x}}_r - K\mathbf{x}_r$$
$$\mathbf{x}_d(0) = \mathbf{x}_r(0), \qquad \dot{\mathbf{x}}_d(0) = \dot{\mathbf{x}}_r(0) \qquad (5)$$

obtained from the impedance model (4), where P_m is the force/torque sensor measurement of the contact force $\mathbf{P}$. Eq. (5) may be interpreted as defining a simple filter that, given definitions for $\mathbf{x}_r(t), M, C$, and K and online measurements $\mathbf{P}_m$, characterizes the desired dynamic relationship between end-effector position and contact force through the specification of $\mathbf{x}_d$. The function of the task space impedance controller is to ensure that the manipulator dynamics (3) evolves in such a way that the end-effector position $\mathbf{x}$ closely tracks $\mathbf{x}_d$ defined in (5).

Let $\dot{\mathbf{x}}_d^* = \dot{\mathbf{x}}_d + \lambda\mathbf{e}$ and $\mathbf{q} = \dot{\mathbf{e}} + \lambda\mathbf{e} = \dot{\mathbf{x}}_d^* - \dot{\mathbf{x}}$ denote the "modified velocity" and weighted position-velocity error, respectively, where $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ is the trajectory tracking error and λ is a positive scalar constant. Consider the following adaptive controller:

$$\mathbf{F} = A(t)\ddot{\mathbf{x}}_d^* + B(t)\dot{\mathbf{x}}_d^* + \mathbf{f}(t) + \mathbf{P}_m + [2k + K(t)]\mathbf{q} \quad (6)$$

where k is a positive scalar constant and $\mathbf{f}(t) \in \Re^n$, $A(t) \in \Re^{n\times n}$, $B(t) \in \Re^{n\times n}$, $K(t) \in \Re^{n\times n}$ are adaptive gains whose update laws are defined to be

$$\begin{aligned} \dot{\mathbf{f}} &= -\alpha_1\mathbf{f} + \beta_1\mathbf{q} \\ \dot{A} &= -\alpha_2 A + \beta_2\mathbf{q}(\ddot{\mathbf{x}}_d^*)^T \\ \dot{B} &= -\alpha_3 B + \beta_3\mathbf{q}(\dot{\mathbf{x}}_d^*)^T \\ \dot{K} &= -\alpha_4 K + \beta_4\mathbf{q}\mathbf{q}^T \end{aligned} \quad (7)$$

In (7), the β_i are positive scalar constants and the α_i are functions of the form $\alpha_i = \alpha_{i0} + \alpha_{i1} \| \dot{\mathbf{x}} \|$, where α_{i0}, α_{i1} are positive scalar constants. Applying the control law (6) to the manipulator dynamics (3) yields the tracking error dynamics

$$H\dot{\mathbf{q}}+V_{cc}\mathbf{q}+2k\mathbf{q}+\mathbf{\Phi}_f+\Phi_A\ddot{\mathbf{x}}_d^*+\Phi_B\dot{\mathbf{x}}_d^*+\Phi_K\mathbf{q}-\mathbf{d}^* = \mathbf{0}, \quad (8)$$

where $\mathbf{\Phi}_f = \mathbf{f} - \mathbf{G}$, $\Phi_A = A - H$, $\Phi_B = B - V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)$, $\Phi_K = K + V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)$, and $\mathbf{d}^* = \mathbf{d} + \mathbf{P} - \mathbf{P}_m$. Note that in obtaining (8) the Coriolis/centripetal term was expanded as follows: $V_{cc}(\mathbf{x}, \dot{\mathbf{x}})\dot{\mathbf{x}}_d^* = V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)\dot{\mathbf{x}} = V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)\dot{\mathbf{x}}_d^* - V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)\mathbf{q}$. Observe that $\mathbf{e}, \dot{\mathbf{e}}$ and $\mathbf{e}, \mathbf{q}$ are diffeomorphic, so that either pair could be combined with $\mathbf{\Phi}_f, \Phi_A, \Phi_B, \Phi_K$ to provide a valid representation for the state of the closed-loop dynamics (7), (8). Indeed, conclusions concerning the evolution of the system in terms of the two sets of coordinates are trivial to relate, because the representations have coincident origins and are linearly related. In what follows, the set of coordinates containing the pair $\mathbf{e}, \mathbf{q}$ will be used as the state variables for the system.

The stability properties of the proposed adaptive controller (6), (7) are summarized in the following theorem.

THEOREM 1. The adaptive controller (6), (7) ensures that $\mathbf{e}, \mathbf{q}, \mathbf{\Phi}_f, \Phi_A, \Phi_B, \Phi_K$ are globally uniformly bounded for all bounded disturbances $\mathbf{d}$. Moreover, if there are no external disturbances or force measurement errors (so that $\mathbf{d} = \mathbf{0}$ and $\mathbf{P}_m = \mathbf{P}$), then the state error $\mathbf{e}, \mathbf{q}$ is guaranteed to converge exponentially to a compact set that can be made arbitrarily small.

Proof. The proof is given in Appendix A. □

Several observations can be made concerning the adaptive control strategy (6), (7). First note that the proposed control law is very simple and requires virtually no information concerning the robot dynamics, and therefore, it provides a computationally efficient, modular, and readily implementable approach to manipulator impedance control. Next, consider the situation where there are no external disturbances or force measurement errors, so that $\mathbf{d}^* \equiv \mathbf{0}$. Theorem 1 states that in this case the tracking errors converge exponentially to a compact set that can be made arbitrarily small by increasing the adaptation gains β_i. Note that exponential convergence ensures that the transient behavior of the errors will be well behaved. In general, when disturbances and force measurement errors are present (so that $\mathbf{d}^*$ is nonzero), the size of the controller errors is influenced by the magnitude of $\mathbf{d}^*$; however, increasing the controller gains can reduce this effect and exponential convergence of the tracking errors to a compact set is still guaranteed.

3.2. Simulation Results

The adaptive impedance control scheme described in Section 3.1 is now applied to a large industrial robot through computer simulation. The robot chosen for this simulation study is the dexterous 7-DOF Robotics Research Corporation (RRC) Model K-1207 arm depicted in Figure 1. The simulation environment incorporates models of all

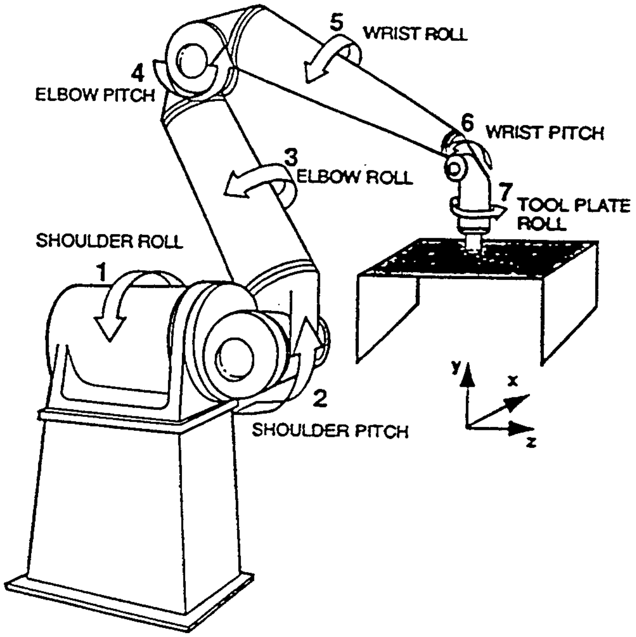


Fig. 1. Illustration of RRC Model K-1207 robot in contact with a surface.

important dynamic subsystems and phenomena, such as full nonlinear arm dynamics, joint stiction, sensor noise, and transmission effects, and therefore provides the basis for a realistic evaluation of controller performance. The control law is applied to the manipulator model with a sampling period of 2 ms, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time step of 2 ms.

This simulation study illustrates how the proposed impedance control scheme can accommodate the transition between unconstrained and constrained end-effector motion and demonstrates that proper utilization of robot redundancy can improve the performance of this transition. In this study a frictionless reaction surface with a stiffness of 100 lb/in is placed in the robot workspace so that it is oriented normal to the y axis and is located at $y = 0.0$. The initial robot configuration is defined to be $\boldsymbol{\theta}(0) = [0°, 0°, 0°, -90°, 0°, -90°, 0°]^T$, and the manipulator is initially at rest. The reference trajectory for end-effector translation is defined as $x_r(t) = 19.675 + 10(1 - \cos \pi t/3), y_r(t) = -26.0 + 13.5(1 - \cos 2\pi t/3), z_r(t) = 0.0$ for $t \in [0, 1.5]$ and $x_r(t) = 19.675 + 10(1 - \cos\pi t/3), y_r(t) = 1.0, z_r(t) = 0.0$ for $t \in [1.5, 3.0]$. In addition, the manipulator is required to maintain its initial end-effector orientation. Note that the reference trajectory intersects the reaction surface at approximately $t = 1.4$, so that at this point the robot undergoes an abrupt transition from unconstrained to constrained motion.

The RRC manipulator has an $n = 7$–dimensional joint space and an $m = 6$–dimensional task space; thus, this robot is redundant for the task of end-effector positioning and possesses one degree of redundancy. In this simulation study, the available redundancy is utilized to reduce the impact force by maximizing the objective function $G(\boldsymbol{\theta}) = \hat{\mathbf{n}}^T H^{-1} \hat{\mathbf{n}}$ at the point of impact, where $\hat{\mathbf{n}}$ is the unit normal to the surface (Walker 1990). The adaptive impedance controller proposed in Section 3.1 is used in this study. The impedance parameters in (5) are defined as $M = \text{diag}[1], K = \text{diag}[10, 5, 10, 10, 10, 10]$, and $C = \text{diag}[2\sqrt{m_{ii}k_{ii}}]$. The adaptive gains $\mathbf{f}, A, B, K$ are set to zero initially, and the remaining controller parameters are set as follows: $k = 20$, $\lambda = 2$, $\alpha_{i0} = \alpha_{i1} = 0.001$, $\beta_i = 10$ for $i = 1, 2, \ldots, 4$. The control input $\mathbf{F}$ is mapped to a vector of joint torques $\mathbf{T}$ using the relation $\mathbf{T} = J_a{}^T \mathbf{F}$. The results of this simulation are given in Figure 2A–2C. Figure 2A indicates that the end-effector closely tracks the reference trajectory in the y direction until contact with the reaction surface is made. At contact, the robot smoothly ceases to track the y component of the reference trajectory in favor of complying with the reaction surface; note from Figure 2B that this behavior has very little effect on the robot motion in the x direction. It can be seen from Figure 2C that the impact force incurred during the transition from unconstrained to constrained motion is acceptable.

4. Adaptive Admittance Controller

We now turn to the development of the proposed adaptive admittance controller for position-controlled manipulators. This approach is motivated by the fact that industrial robots are always equipped with a position control system and addresses the problem of adaptively specifying position setpoints for this inner-loop controller so that the desired compliant behavior is realized. Thus, in this method of compliant motion, the manipulator and position controller are viewed together as a single system to be controlled using position commands as the control inputs. This approach has the advantage of being readily implementable with industrial robots. In addition, because this is a force control methodology, it is appropriate for those applications that require accurate regulation of the end-effector/environment contact force despite uncertainty regarding the location and stiffness of the environment.

In what follows, the inner-loop position controller is assumed to have a proportional-derivative (PD) feedback structure, as this is a simple and well-known strategy and is representative of existing industrial robot controllers; however, the analysis techniques utilized here are applicable to other inner-loop controllers as well. Let $\mathbf{x}_d \in \mathcal{X}$ define the desired task space trajectory and $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ denote the trajectory tracking error. Consider the PD control law

$$\mathbf{F} = K_1 \mathbf{e} - K_2 \dot{\mathbf{x}}, \tag{9}$$

where $K_1, K_2 \in \Re^{n \times n}$ are constant feedback matrices. Applying the control law (9) to the manipulator dynamics (3) yields

$$H\ddot{\mathbf{x}} + V_{cc}\dot{\mathbf{x}} + k_2\dot{\mathbf{x}} + k_1\mathbf{x} + \mathbf{G} + \mathbf{d} + \mathbf{P} = k_1\mathbf{x}_d, \tag{10}$$

where, for simplicity of development, the feedback matrices are assumed to have the form $K_1 = \text{diag}[k_1], K_2 = \text{diag}[k_2]$, with k_1, k_2 positive scalar constants. The differential equation (10) quantifies the dynamics of the PD controlled manipulator.

4.1. Control Scheme

Consider the position-controlled manipulator (10) interacting with a linearly elastic environment. Let $\mathbf{x}_c$ represent the p generalized coordinates to be under compliant motion control and $\mathbf{x}_p$ be the $(n-p)$ position-controlled coordinates. Assume that $\mathbf{x}$ is ordered so that $\mathbf{x} = [\mathbf{x}_p^T \;\; \mathbf{x}_c^T]^T$, and note that this decomposition can always be achieved in the constraint frame by using a properly defined coordinate transformation (e.g., Lozano and Brogliato 1992).

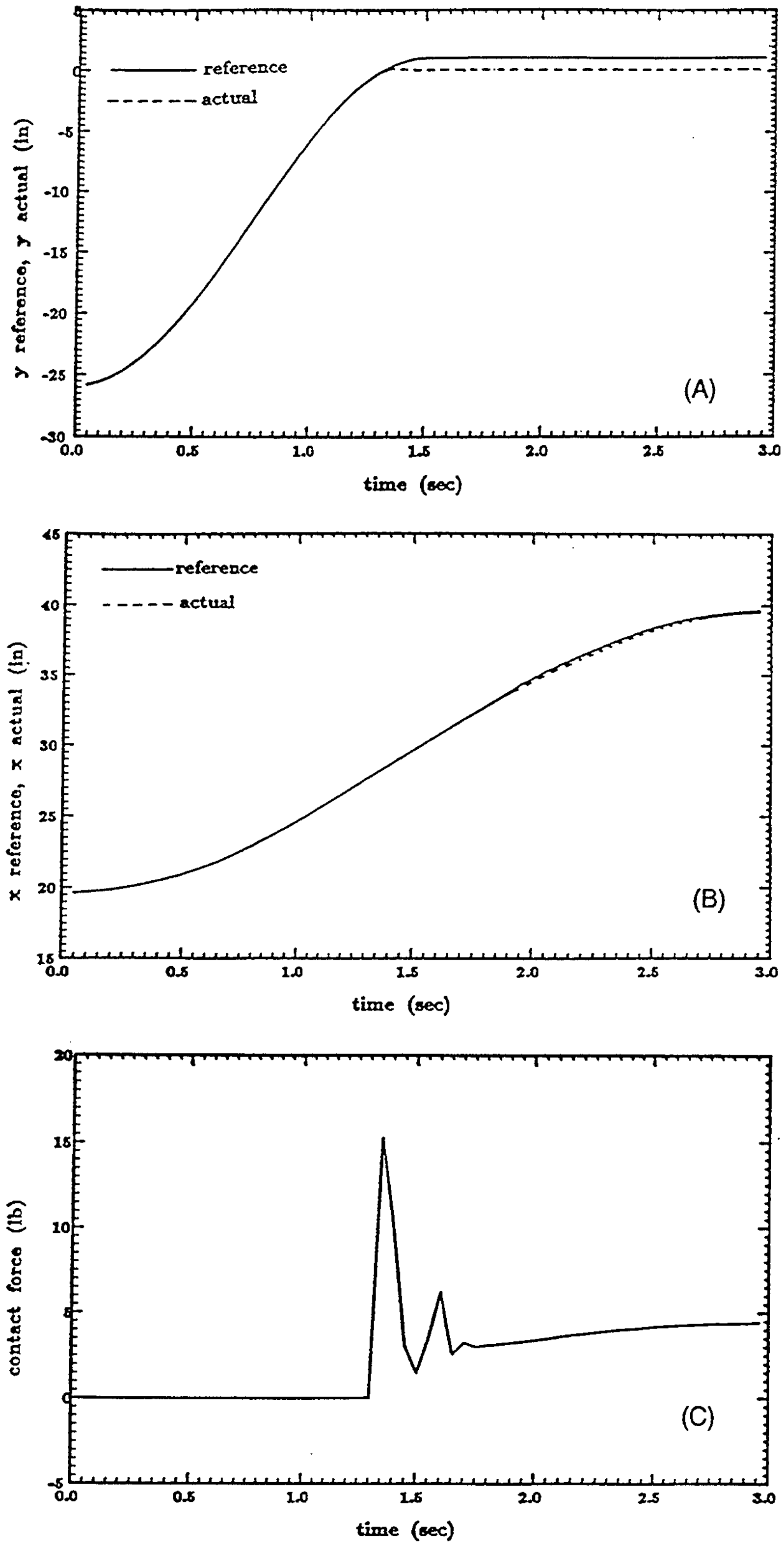


Fig. 2. A, Response of end-effector coordinate y in impedance control simulation example. B, Response of end-effector coordinate x in impedance control simulation example. C, End-effector/environment contact force P in impedance control simulation example.

The manipulator/environment interaction force is then given by $\mathbf{P} = [\mathbf{0}^T \;\; \mathbf{P}_c^T]^T$, where $\mathbf{P}_c \in \Re^p$ has the form

$$\mathbf{P}_c = K_{\text{env}}(\mathbf{x}_c - \mathbf{x}_{\text{env}}), \tag{11}$$

where $K_{\text{env}} \in \Re^{p \times p}$ and $\mathbf{x}_{\text{env}} \in \Re^p$ are the environmental stiffness and location, respectively, and $K_{\text{env}} = \text{diag}\,[k_{\text{env}}]$ is assumed for simplicity of presentation.

Let $\mathbf{x}_{pd}$ and $\mathbf{P}_d$ represent the desired constant end-effector position and contact force for the manipulator, respectively. The adaptive admittance control problem involves designing a strategy for specifying $\mathbf{x}_d(t)$ so that $\mathbf{x}_p$, $\mathbf{P}_c$ approach $\mathbf{x}_{pd}$, $\mathbf{P}_d$ despite considerable uncertainty regarding the manipulator, the inner-loop position controller, and the environment. Implicit in this problem definition is the objective of utilizing the admittance controller not only to provide compliance control (in the force subspace), but also to improve the accuracy of the inner-loop position controller (in the position subspace). Consider the control law

$$\mathbf{x}_d = \begin{bmatrix} \mathbf{x}_{pd} + \mathbf{f}_p(t) + K_p(t)(\mathbf{x}_{pd} - \mathbf{x}_p) \\ \mathbf{f}_c(t) + K_c(t)\mathbf{e}_c \end{bmatrix}, \tag{12}$$

where $\mathbf{f}_p \in \Re^{n-p)}$, $\mathbf{f}_c \in \Re^p$, $K_p \in \Re^{(n-p)\times(n-p)}$, $K_c \in \Re^{p \times p}$ are adaptive gains and $\mathbf{e}_c = \mathbf{P}_d - \mathbf{P}_c$. Applying the control strategy (12) to the position-controlled manipulator dynamics (10) yields the error dynamics

$$H\ddot{\mathbf{e}}_f + V_{cc}\dot{\mathbf{e}}_f - k_2\dot{\mathbf{e}}_f + k_1\mathbf{e}_f + \begin{bmatrix} \phi_p \\ \phi_c \end{bmatrix} \tag{13}$$
$$+ \begin{bmatrix} o_{kp} & 0 \\ 0 & \phi_{kc} \end{bmatrix} \mathbf{e}_f - k_{\text{env}}\mathbf{d} = \mathbf{0},$$

where $\mathbf{e}_f = [k_{\text{env}}(\mathbf{x}_{pd} - \mathbf{x}_p)^T \mid (\mathbf{P}_d - \mathbf{P}_c)^T]^T$, $\phi_p = k_1 k_{\text{env}}\mathbf{f}_p - k_{\text{env}}\mathbf{G}_p$, $\phi_c = k_1 k_{\text{env}}\mathbf{f}_c - k_1 k_{\text{env}}\mathbf{x}_{\text{env}} - (k_1 + k_{\text{env}})\mathbf{P}_d - k_{\text{env}}\mathbf{G}_c$, $\phi_{kp} = k_1 K_p$, $o_{kc} = k_1 k_{\text{env}} K_c + k_{\text{env}} I$, I is the $p \times p$ identity matrix, and the gravity vector is partitioned as $\mathbf{G} = [\mathbf{G}_p^T \;\; \mathbf{G}_c^T]^T$.

Observe that $\mathbf{e}_f = \dot{\mathbf{e}}_f = \mathbf{0}$, $\phi_p = \phi_c = \mathbf{0}$, $\phi_{kp} = \phi_{kc} = 0$ is an equilibrium point for (13) in the absence of external disturbances (so that $\mathbf{d} = \mathbf{0}$), and this equilibrium point corresponds to $\mathbf{x}_p = \mathbf{x}_{pd}$, $\mathbf{P}_c = \mathbf{P}_d$ as desired. Thus, the control objective is to choose update laws for $\mathbf{f} = [\mathbf{f}_p^T \;\; \mathbf{f}_c^T]^T$ and $K = [K_p^T \;\; K_c^T]^T$ that ensure that this equilibrium is globally stable and that a (small) neighborhood of this equilibrium is globally attractive. Toward this end, define the following adaptation laws for $\mathbf{f}$ and K:

$$\dot{\mathbf{f}} = -\alpha_1\mathbf{f} + \beta_1\mathbf{q}, \qquad \dot{K} = -\alpha_2 K + \beta_2\mathbf{q}\mathbf{e}_f^T \tag{14}$$

where $\mathbf{q} = \dot{\mathbf{e}}_f - \epsilon\mathbf{e}_f$, ϵ is a positive constant, the β_i are positive scalar constants, and the α_i are functions of the form $\alpha_i = \alpha_{i0} + \alpha_{i1}\| \dot{\mathbf{x}} \|$ with α_{i0}, α_{i1} positive scalar constants.

The following theorem indicates that the proposed adaptive admittance controller (12), (14) realizes the control objective of accurate position regulation in the position subspace and force regulation in the force subspace.

THEOREM 2. Suppose that the controller parameter ϵ is chosen so that

$$\epsilon < \min\left[\frac{(k_1\lambda_{\min}(H))^{1/2}}{\lambda_{\max}(H)}, \frac{k_2}{k_2^2/4k_1 + \lambda_{\max}(H) + e_{\max}k_{cc}}\right] \tag{15}$$

where k_{cc} satisfies $\| V_{cc} \|_F \leq k_{cc} \| \dot{\mathbf{e}}_f \|$ (this bound exists as a direct consequence of Lemma 1) and $e_{\max}$ is an upper bound on the task space position error. (This bound exists because $\mathcal{N}$ is compact and $\mathbf{h}_a$ is continuous; see also Lozano and Brogliato [1992], Wen et al. [1992], and Colbaugh et al. [1994].) Then the adaptive controller (12), (14) ensures that $\mathbf{e}_f$, $\dot{\mathbf{e}}_f$, ϕ_p, ϕ_c, ϕ_{kp}, ϕ_{kc} are globally uniformly bounded for all bounded disturbances $\mathbf{d}$. Moreover, if there are no external disturbances (so that $\mathbf{d} = \mathbf{0}$), then $\mathbf{e}_f, \dot{\mathbf{e}}_f$ are guaranteed to converge exponentially to a compact set that can be made arbitrarily small.

Proof. The proof is given in Appendix B. □

Several observations can be made concerning the adaptive admittance control strategy (12), (14). First, note that the proposed scheme is extremely simple and requires very little information concerning the manipulator model, the inner-loop position controller, or the environment. For example, the controller parameter selection process requires only very conservative estimates for manipulator model terms. The adaptive strategy (12), (14) ensures that in the absence of external disturbances, the controller errors converge exponentially to a compact set that can be made arbitrarily small (by increasing the adaptation gains β_i). If disturbances are present (so that $\mathbf{d}$ is nonzero), the size of the controller errors is influenced by the magnitude of $\mathbf{d}$; however, increasing the controller gains can reduce this effect. Therefore, the proposed controller provides a computationally efficient and modular solution to the manipulator compliant motion control problem. The fact that the control scheme is designed to be used with position-controlled manipulators makes implementation with industrial robots particularly convenient. Note that the weighted error $\mathbf{q}$ used in the adaptive laws (14) contains $\dot{\mathbf{e}}_c$, suggesting that implementation of the proposed controllers requires $\dot{\mathbf{P}}_c$, which is ordinarily not available. It is easy to show by direct integration that, although $\dot{\mathbf{f}}$ and $\dot{K}$ depend on $\dot{\mathbf{e}}_c$, $\mathbf{f}$ and K do not, so that the controller does not require force derivative information for implementation.

4.2. Experimental Results

The adaptive admittance controller (12), (14) is now applied to an industrial robot in a series of experiments. The experimental facility utilized for this study was the Jet Propulsion Laboratory Supervisory Telerobotics Research Laboratory, which consists of a dexterous 7-DOF RRC K-1207 redundant manipulator and controller, a 6-DOF Lord wrist force/torque sensor, a TRI servoed gripper, a VME-based real-time computing environment with six Motorola 68020 processor boards, a Sun 4/260 workstation, and a Silicon Graphics IRIS workstation. This laboratory is designed to be a general space telerobotics research and development facility. One of the consequences of this focus is that, in its present configuration, the update rate for any outer-loop control strategy that commands task space position setpoints can be no greater than 35 Hz. Note that this slow update rate presents a severe challenge to any adaptive outer-loop control algorithm.

The adaptive admittance control scheme (12), (14) is implemented in the present experimental study. The task specified in the experiment requires the manipulator to approach the environment, accommodate the transition from unconstrained to constrained motion, and then exert a user-defined contact force on the environment. Two different environments are utilized in the experiments: a "soft" environment consisting of books stacked on a cardboard box, and a "stiff" environment obtained by mounting a metal plate on a steel table. No a priori knowledge of the manipulator model, the inner-loop position controller, or the environment location or stiffness is provided to the outer-loop admittance control strategy. In each experiment the reaction surface is placed in the robot workspace so that it is oriented normal to the y axis and is located approximately 0.25 m from the initial end-effector position. The manipulator is commanded to exert a contact force of 10.0 N on the reaction surface and to maintain its initial end-effector x and z positions and orientation; note that with this task definition the manipulator approaches the reaction surface under the control of the admittance control scheme and with the objective of reducing the contact force error. The available manipulator redundancy is utilized to reduce the impact force using the strategy summarized in Section 3.2.

The experiments are designed to evaluate the effectiveness of the admittance controller (12), (14) for establishing stable contact and for achieving accurate force regulation in the presence of considerable uncertainty regarding the manipulator, inner-loop position controller, and environment. In addition, these experiments examine the sensitivity of the controller to a large variation in environmental stiffness. For all experiments, the adaptive gains $\mathbf{f}, K$ are initialized to zero, ϵ is set equal to 1, and the adaptation gains are chosen to be $\beta_1 = 0.01$ and $\beta_2 = 0.0001$ (for all gains, the units for force and length are Newton and centimeter). The results of these experiments are given in Figures 3 and 4, with Figure 3 showing the response of the contact force for the soft environment and Figure 4 illustrating the response for the stiff environment. It can be seen from the figures that in both cases stable contact is rapidly established, and accurate force set point regulation is quickly achieved, despite the lack of a priori information and the very slow update rate (35 Hz). In addition, it is observed that the performance is robust to large variations in environmental characteristics (cardboard box and steel plate). It should be noted that in the present experimental setup, gravity loading on the force/torque sensor is only approximately compensated with calibration, so that forces with magnitudes less than 2.0 N are ignored in the software: this accounts for the offset that appears in the plots of contact force before contact with the environment is made.

5. Conclusions

This article presents two new adaptive compliant motion control schemes for rigid-link manipulators. The effectiveness of the proposed schemes is illustrated through both computer simulations and laboratory experiments with a dexterous RRC 7-DOF manipulator. These studies indicate that the adaptive impedance controller provides an accurate and robust means of achieving the desired end-effector impedance characteristics and that the adaptive admittance scheme represents a readily implementable method of obtaining high performance force control. Furthermore, the investigations demonstrate that manipulator redundancy can be used to improve the performance of compliance control tasks involving abrupt transition from unconstrained to constrained end-effector motion by reducing the impact force.

Appendix A: Proof of Theorem 1

In this appendix the proof of Theorem 1 is sketched: a detailed proof of the theorem is given in Colbaugh et al. (1993a). Consider the Lyapunov function candidate

$$V = \frac{1}{2}\mathbf{q}^T H \mathbf{q} + k\lambda \mathbf{e}^T \mathbf{e} + \frac{1}{2\beta_1}\mathbf{\Phi}_f^T \mathbf{\Phi}_f \qquad \text{(A1)}$$
$$+\frac{1}{2}\mathrm{tr}\left[\frac{1}{\beta_2}\Phi_A\Phi_A^T + \frac{1}{\beta_3}\Phi_B\Phi_B^T + \frac{1}{\beta_4}\Phi_K\Phi_K^T\right]$$

and note that V is a positive definite and radially unbounded function of the closed-loop system state. Computing the derivative of (A1) along (7), (8) and simplifying using Lemma 1 yields

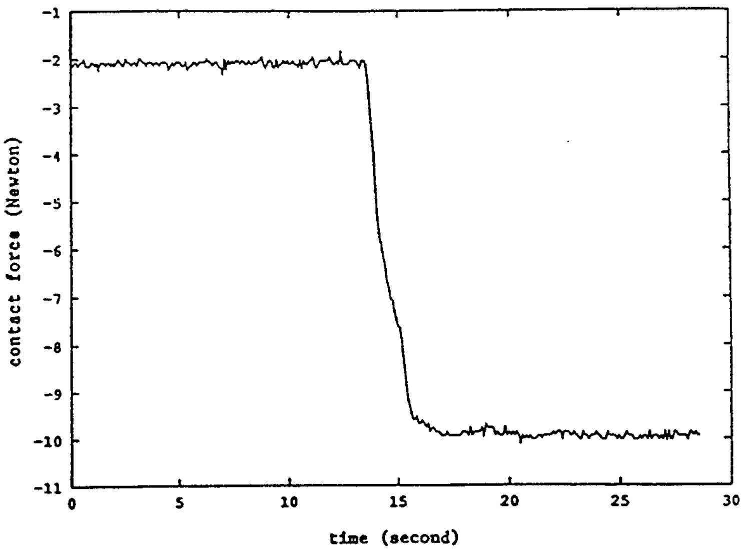


Fig. 3. End-effector/environment contact force in experimental study (soft environment).

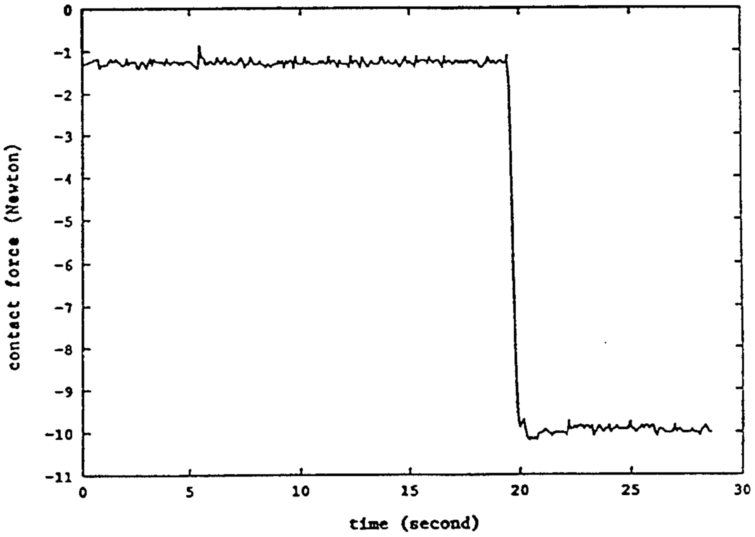


Fig. 4. End-effector/environment contact force in experimental study (stiff environment).

$$
\begin{aligned}
\dot{V} = & -k \parallel \mathbf{q} \parallel^2 - k \parallel \dot{\mathbf{e}} \parallel^2 - k\lambda^2 \parallel \mathbf{e} \parallel^2 + \mathbf{q}^T \mathbf{d}^* \\
& - \frac{\alpha_1}{\beta_1} \parallel \Phi_f - \frac{1}{2\alpha_1}(\dot{\mathbf{G}} + \alpha_1 \mathbf{G}) \parallel^2 - \frac{\alpha_2}{\beta_2} \parallel \Phi_A \\
& + \frac{1}{2\alpha_2}(\dot{H} - \alpha_2 H) \parallel_F^2 - \frac{\alpha_3}{\beta_3} \parallel \Phi_B \\
& + \frac{1}{2\alpha_3}(\dot{V}_{cc} - \alpha_3 V_{cc}) \parallel_F^2 - \frac{\alpha_4}{\beta_4} \parallel \Phi_K \qquad \text{(A2)} \\
& - \frac{1}{2\alpha_4}(\dot{V}_{cc} - \alpha_4 V_{cc}) \parallel_F^2 + \frac{1}{4\alpha_1\beta_1} \parallel \dot{\mathbf{G}} + \alpha_1 \mathbf{G} \parallel^2 \\
& + \frac{1}{4\alpha_2\beta_2} \parallel \dot{H} + \alpha_2 H \parallel_F^2 + \frac{1}{4\alpha_3\beta_3} \parallel \dot{V}_{cc} + \alpha_3 V_{cc} \parallel_F^2 \\
& + \frac{1}{4\alpha_4\beta_4} \parallel \dot{V}_{cc} + \alpha_4 V_{cc} \parallel_F^2
\end{aligned}
$$

where $\parallel \cdot \parallel_F$ is the Frobenius norm and $V_{cc} = V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)$. The properties of $H, V_{cc}, \mathbf{G}$ summarized in Lemma 1. together with the boundedness of $\mathbf{x}_r$, $\dot{\mathbf{x}}_r$, and $\ddot{\mathbf{x}}_r$, permit the following bound on $\dot{V}$ in (A2) to be derived:

$$
\begin{aligned}
\dot{V} \leq & -\frac{k}{4} \parallel \mathbf{q} \parallel^2 - \frac{k\lambda^2}{4} \parallel \mathbf{e} \parallel^2 - \frac{\alpha_{10}}{4\beta_1} \parallel \mathbf{\Phi}_f \parallel^2 \\
& - \frac{\alpha_{20}}{4\beta_2} \parallel \Phi_A \parallel_F^2 - \frac{\alpha_{30}}{4\beta_3} \parallel \Phi_B \parallel_F^2 - \frac{\alpha_{40}}{4\beta_4} \parallel \Phi_K \parallel_F^2 \\
& + d_{\max}\delta_0 + \frac{1}{\beta_1}\delta_1 + \frac{1}{\beta_2}\delta_2 + \frac{1}{\beta_3}\delta_3 + \frac{1}{\beta_4}\delta_4 \qquad \text{(A3)}
\end{aligned}
$$

where $d_{\max}$ is an upper bound on $\parallel \mathbf{d}^* \parallel$ and the δ_i are positive scalar constants that do not increase as the β_i increase. In obtaining these bounds the compactness of

$\mathcal{N}$ and continuity of $\mathbf{h}_a$ are utilized to show that $\mathbf{x}$ and $\mathbf{P}$ are bounded, and therefore $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$, and $\dot{\mathbf{x}}_d^*$ are bounded. Then, using Lemma 1, it can be concluded that $V_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)$ is bounded and $\dot{V}_{cc}(\mathbf{x}, \dot{\mathbf{x}}_d^*)$ grows linearly in $\dot{\mathbf{e}}$. Arguments of this type concerning the implications of a compact configuration manifold on the possible magnitude of the configuration error have also been used in Lozano and Brogliato (1992), Wen et al. (1992), and Colbaugh et al. (1994).

Let $\mathbf{z} = [\| \mathbf{e} \| \;\; \| \mathbf{q} \|]^T$, $\mathbf{\Phi} = [\| \Phi_f \| \;\; \| \Phi_A \|_F \;\; \| \Phi_B \|_F \;\; \| \Phi_K \|_F]^T$, $\beta_{\min} = \min(\beta_i)$, and $\beta_{\max} = \max(\beta_i)$. If $\beta_{\max}/\beta_{\min}$ is fixed, then there exist positive constants λ_i independent of the β_i, and positive constants B_i that do not increase as $\beta_{\min}$ increases, such that V and $\dot{V}$ in (A1) and (A3) can be bounded as

$$\lambda_1 \| \mathbf{z} \|^2 + \frac{\lambda_2}{\beta_{\min}} \| \mathbf{\Phi} \|^2 \le V \le \lambda_3 \| \mathbf{z} \|^2 + \frac{\lambda_4}{\beta_{\min}} \| \mathbf{\Phi} \|^2$$

$$\dot{V} \le -\lambda_5 \| \mathbf{z} \|^2 - \frac{\lambda_6}{\beta_{\min}} \| \mathbf{\Phi} \|^2 + \frac{1}{\beta_{\min}} B_1 + d_{\max} B_2$$

Lemma 2 then permits the conclusion that $\| \mathbf{z} \|$, $\| \mathbf{\Phi} \|$ are uniformly bounded, which implies that $\mathbf{e}$, $\mathbf{q}$, $\Phi_f, \Phi_A, \Phi_B, \Phi_K$ are uniformly bounded. Moreover, in the case of no external disturbances or force measurement errors (so that $d_{\max} = 0$), $\| \mathbf{z} \|$ and $\| \mathbf{\Phi} \|$ converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where $r_1 = (\delta B_1/\lambda_1 \beta_{\min})^{1/2}$, $r_2 = (\delta B_1/\lambda_2)^{1/2}$, and $\delta = \max(\lambda_3/\lambda_5, \lambda_4/\lambda_6)$. Observe that the radius of the ball to which $\| \mathbf{z} \|^2 = \| \mathbf{e} \|^2 + \| \mathbf{q} \|^2$ is guaranteed to converge can be decreased as desired simply by increasing $\beta_{\min}$.

Appendix B: Proof of Theorem 2

In this appendix, the proof of Theorem 2 is summarized; additional details can be found in Colbaugh et al. (1993a). In what follows, let $\phi = [\phi_p^T \;\; \phi_c^T]^T$ and $\phi_K = \text{block diag}[\phi_{kp}, \phi_{kc}]$ for notational convenience. Consider the Lyapunov function candidate

$$V = \frac{1}{2}\dot{\mathbf{e}}_f^T H \dot{\mathbf{e}}_f + \frac{1}{2} k_1 \mathbf{e}_f^T \mathbf{e}_f + \epsilon \mathbf{e}_f^T H \dot{\mathbf{e}}_f + \frac{1}{2\beta_1 k_1 k_{\text{env}}} \phi^T \phi + \frac{1}{2\beta_2 k_1 k_{\text{env}}} \text{tr}[\phi_K \phi_K^T] \tag{B1}$$

and observe that V is positive definite and radially unbounded in all arguments, provided $\beta_1, \beta_2, \epsilon$ are positive scalar constants and ϵ is chosen so that the inequality (15) is satisfied. Computing the derivative of (B1) along (13), (14) and simplifying using Lemma 1 yields

$$\begin{aligned}
\dot{V} = & -k_2 \| \dot{\mathbf{e}}_f \|^2 - \epsilon k_1 \| \mathbf{e}_f \|^2 - \epsilon \dot{\mathbf{e}}_f^T H \dot{\mathbf{e}}_f \\
& + \epsilon \mathbf{e}_f^T [V_{cc}^T \dot{\mathbf{e}}_f - k_2 \dot{\mathbf{e}}_f] + \frac{1}{\beta_1 k_1 k_{\text{env}}} \phi^T \dot{\phi} \\
& + \frac{1}{\beta_2 k_1 k_{\text{env}}} \text{tr}[\phi_K \dot{\phi}_K^T] + k_{\text{env}} \mathbf{q}^T \mathbf{d} \\
& - \mathbf{q}^T \phi - \mathbf{q}^T \phi_K \mathbf{e}_f \\
\le & -\frac{1}{4} [\| \mathbf{e}_f \| \;\; \| \dot{\mathbf{e}}_f \|] Q \begin{bmatrix} \| \mathbf{e}_f \| \\ \| \dot{\mathbf{e}}_f \| \end{bmatrix} \\
& - \frac{\alpha_{10}}{4\beta_1 k_1 k_{\text{env}}} \| \phi \|^2 - \frac{\alpha_{20}}{4\beta_2 k_1 k_{\text{env}}} \| \phi_K \|_F^2 \\
& + d_{\max} \delta_5 + \frac{1}{\beta_1} \delta_6 + \frac{1}{\beta_2} \delta_7
\end{aligned} \tag{B2}$$

where the δ_i are positive scalar constants that do not increase as β_1, β_2 increase, the matrix Q is given by

$$Q = \begin{bmatrix} \epsilon k_1 & -\frac{1}{2}\epsilon k_2 \\ -\frac{1}{2}\epsilon k_2 & k_2 - \epsilon \lambda_{\max}(H) - \epsilon \epsilon_{\max} k_{cc} \end{bmatrix}$$

and the upper bound (B2) was obtained using the procedure adopted in deriving the bound (A3) in the proof of Theorem 1. Note that if ϵ is chosen so that (15) is satisfied, then Q is positive definite. Let $\mathbf{z} = [\| \mathbf{e}_f \| \;\; \| \dot{\mathbf{e}}_f \|]^T$, $\mathbf{\Phi} = [\| \phi \| \;\; \| \phi_K \|_F]^T$, $\beta_{\min} = \min(\beta_1, \beta_2)$, and $\beta_{\max} = \max(\beta_1, \beta_2)$. If $\beta_{\max}/\beta_{\min}$ is fixed, then there exist positive scalar constants λ_i independent of β_1, β_2, and positive constants B_3, B_4 that do not increase with β_1, β_2, such that V and $\dot{V}$ in (B1) and (B2) can be bounded as

$$\lambda_1 \| \mathbf{z} \|^2 + \frac{\lambda_2}{\beta_{\min}} \| \mathbf{\Phi} \|^2 \le V \le \lambda_3 \| \mathbf{z} \|^2 - \frac{\lambda_4}{\beta_{\min}} \| \mathbf{\Phi} \|^2$$

$$\dot{V} \le -\lambda_5 \| \mathbf{z} \|^2 - \frac{\lambda_6}{\beta_{\min}} \| \mathbf{\Phi} \|^2 + \frac{1}{\beta_{\min}} B_3 + d_{\max} B_4$$

Lemma 2 then permits the conclusion that $\| \mathbf{z} \|$, $\| \mathbf{\Phi} \|$ are uniformly bounded, which implies that $\mathbf{e}_f$, $\dot{\mathbf{e}}_f$, ϕ, ϕ_K are uniformly bounded. Moreover, in the case of no external disturbances (so that $d_{\max} = 0$), $\| \mathbf{z} \|$ and $\| \mathbf{\Phi} \|$ converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where $r_1 = (\delta B_3/\lambda_1 \beta_{\min})^{1/2}$, $r_2 = (\delta B_3/\lambda_2)^{1/2}$, and $\delta = \max(\lambda_3/\lambda_5, \lambda_4/\lambda_6)$. Observe that the radius of the ball to which $\| \mathbf{z} \|^2 = \| \mathbf{e}_f \|^2 + \| \dot{\mathbf{e}}_f \|^2$ is guaranteed to converge can be decreased as desired simply by increasing $\beta_{\min}$.

Acknowledgments

The research described in this article was supported through contracts with the Jet Propulsion Laboratory, the National Aeronautics and Space Administration, the Army Research Office, and the Department of Energy Waste-management Education and Research Consortium (WERC).

References

Carelli, R., and Kelly, R. 1991. An adaptive impedance/force controller for robot manipulators. *IEEE Trans. Auto. Control* 36(8):967–971.

Colbaugh, R., Seraji, H., and Glass, K. 1993a (March). Adaptive compliant motion control of manipulators. Robotics Laboratory report 93-01. Las Cruces, NM, New Mexico State University.

Colbaugh, R., Seraji, H., and Glass, K. 1993b. Direct adaptive impedance control of robot manipulators. *J. Robot. Sys.* 10(2):217–248.

Colbaugh, R., Seraji, H., and Glass, K. 1994. Decentralized adaptive control of manipulators. *J. Robot. Sys.* 11(5):425–440.

Corless, M. 1990. Guaranteed rates of exponential convergence for uncertain systems. *J. Optim. Theory Appl.* 64(3):481–494.

Hogan, N. 1985. Impedance control: An approach to manipulation. parts I–III. *ASME J. Dynam. Sys. Measurement Control* 107(1):1–24.

Jean, J., and Fu, L. 1993. Adaptive hybrid control strategies for constrained robots. *IEEE Trans. Auto. Control* 38(4):598–603.

Lozano, R., and Brogliato, B. 1992. Adaptive hybrid force-position control for redundant manipulators. *IEEE Trans. Auto. Control* 37(10):1501–1505.

Lu, W., and Meng, Q. 1991. Impedance control with adaptation for robotic manipulators. *IEEE Trans. Robot. Auto.* 7(3):408–415.

Niemeyer, G., and Slotine, J. J. 1991. Performance in adaptive manipulator control. *Int. J. Robot. Res.* 10(2):149–161.

Raibert, M., and Craig, J. 1981. Hybrid position/force control of manipulators. *ASME J. Dynam. Sys. Measurement Control* 102(2):126–133.

Reed, J., and Ioannou, P. 1989. Instability analysis and robust adaptive control of robotic manipulators. *IEEE Trans. Robot. Auto.* 5(3):381–386.

Seraji, H. 1989. Configuration control of redundant manipulators: Theory and implementation. *IEEE Trans. Robot. Auto.* 5(4):472–490.

Stepanenko, Y., and Yuan, J. 1992. Robust adaptive control of a class of nonlinear mechanical systems with unbounded and fast-varying uncertainties. *Automatica* 28(2):265–276.

Volpe, R., and Khosla, P. 1993. A theoretical and experimental investigation of explicit force control strategies for manipulators. *IEEE Trans. Auto. Control* 38(11):1634–1650.

Walker, I. 1990 (April, Cincinnati). The use of kinematic redundancy in reducing impact and contact effects in manipulation. *Proc. IEEE Int. Conf. Robot. and Auto.*

Wen, J., Kreutz-Delgado, K., and Bayard, D. 1992. Lyapunov function-based control laws for revolute robot arms: Tracking control, robustness, and adaptive control. *IEEE Trans. Auto. Control* 37(2):231–237.

By using an integration by parts, the coordinates $x(t)$ and $y(t)$ as functions of $\theta(t)$ are

$$\begin{aligned} x(t) &= k_a(\cos\theta - \cos\theta_0 + (\theta - \theta_0 + \tfrac{\omega_0 v_0}{a})\sin\theta \\ &\quad - \tfrac{\omega_0 v_0}{a}\sin\theta_0) + x_0 \\ y(t) &= k_a(\sin\theta - \sin\theta_0 - (\theta - \theta_0 + \tfrac{\omega_0 v_0}{a})\cos\theta \\ &\quad + \tfrac{\omega_0 v_0}{a}\cos\theta_0) + y_0 \\ \theta(t) &= \omega_0 t + \theta_0 \end{aligned}$$

When $v_0 < 0$, there is a point $\gamma(t)$ such that $v(t) = 0$. This is a cusp. The corresponding configuration $\hat{c}$ verifies

$$x_{\hat{c}} = k_a(\cos\theta_{\hat{c}} - \cos\theta_0 + (\theta_{\hat{c}} - \theta_0)\sin\theta_0) + x_0 \tag{11}$$

$$y_{\hat{c}} = k_a(\sin\theta_{\hat{c}} - \sin\theta_0 - (\theta_{\hat{c}} - \theta_0)\cos\theta_0) + y_0 \tag{12}$$

$$\theta_{\hat{c}} = -\frac{v_0\omega_0}{a} + \theta_0 \tag{13}$$

Acknowledgment

The authors are grateful to Matthieu Herrb for his help, especially in producing the graphic interface.

References

[1] J. Iijima, Y. Kanayama, and S. Yuta, "A locomotion control system for mobile robots," in *7th Int. Joint Conf. Artificial Intell. (IJCAI)*, 1981.

[2] H. Chochon and B. Leconte, "Etude d'un module de locomotion pour un robot mobile," in Rapport de fin d'étude ENSAE, Laboratoire d'Automatique et d'Analyse des Systèmes (C.N.R.S.), , 1983.

[3] Y. Kanayama and N. Miyake, "Trajectory generation for mobile robots," in *Robotics Research 3*, G. Giralt and O. Faugeras, Eds. Boston, MA: MIT Press, 1986.

[4] James and James, *Mathematics Dictionary*. Princeton, NJ: Van Nostrand, 1968.

[5] P. Jacobs, A. Rege, and J.-P. Laumond, "Non-holonomic motion planning for hilare-like mobile robot," in *Int. Symp. Intell. Robot. (ISIR)*, 1991.

[6] P. Hartman, "The highway spiral for combining curves of different radii," *Trans. Amer. Soc. Civil Eng.*, 1957.

[7] D. Meek and D. Walton, "Clothoid spline transition spirals," *Math. Computation*, vol. 59, no. 199, July 1992.

[8] F. Noreils, A. Khoumsi, G. Baugil, and R. Chatila, "Reactive processes for mobile robot control," in *4th ICAR '89*, 1989.

[9] D. Shin and S. Singh, "Path generation for robot vehicules using composite clothoid segments," Technical Report CMU-RI-TR-90-31, Robotics Institute, Carnegie Mellon University, 1990.

[10] Y. Kanayama and B. Hartman, "Smooth local planning for autonomous vehicles," in *IEEE Int. Conf. on Robot. and Automat.*, 1989.

[11] H. Delingette, M. Herbert, and K. Ikeuchi, "Trajectory generation with curvature constraint based on energy minimization," in *IEEE Int. Workshop on Intell. Robots and Syst. (IROS '91)*,, 1991.

[12] A. Segovia, M. Rombaut, A. Preciado, and D. Meizel, "Comparative study of the different methods of path generation for a mobile robot in a free environment," in *ICAR '91*, 1991.

[13] R. Chatila, "Mobile robot navigation: space modeling and decisional processes," in *Robotics Research: The Third Int. Symp.*, , 1985.

[14] L. Pontryagin, V. Boltyansky, R. Gamkrelidze, and E. Mischenko, *The Mathematical Theory of Optimal Processes*. New York: Wiley, 1962.

[15] D. Reister and F. Pin, "Time optimal trajectories for mobile robots with two independently driven wheels," *Int. J. Robot. Res.*, vol. 13, 1994.

[16] P. Souères and J. P. Laumond, "Shortest path synthesis for a car-like robot," in *European Cont. Conf.*, 1993.

[17] J. P. Laumond, T. Simeon, R. Chatila, and G. Giralt, "Trajectory planning and motion control for mobile robots," in *Geometry and Robotics*,, J. D. Boissonnat and J. P. Laumond, Eds. (Lecture Notes in Computer Science, Vol. 391). New York: Springer Verlag, 1989, pp. 133–149.

[18] J. P. Laumond, P. E. Jacobs, M. Taïx, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. Robot. and Automat.*, vol. 10, no. 5, pp. 577–593, 1994.

Real-Time Collision Avoidance for Redundant Manipulators

K. Glass, *Member, IEEE*, R. Colbaugh, *Member, IEEE*, D. Lim, and H. Seraji, *Senior Member, IEEE*,

***Abstract*—This paper presents a simple and robust approach to achieving collision avoidance for kinematically redundant manipulators at the control-loop level. The proposed scheme represents the obstacle avoidance requirement as inequality constraints in the manipulator workspace, and ensures that these inequalities are satisfied while the end-effector tracks the desired trajectory. The control scheme is the damped-least-squares formulation of the configuration control approach implemented as a kinematic controller. Computer simulation and experimental results are given for a Robotics Research 7 DOF redundant arm and demonstrate the collision avoidance capability for reaching inside a truss structure. These results confirm that the proposed approach provides a simple and effective method for real-time collision avoidance.**

I. Introduction

In Oct. 1990, the National Aeronautics and Space Administration (NASA) initiated a research and development project on Remote Surface Inspection (RSI) at the Jet Propulsion Laboratory (JPL). The goal of this project is to develop and demonstrate the necessary technologies for the inspection of space structures, such as the Space Station Freedom (SSF), using remote robots for sensor placement under supervisory control. The purpose of the inspection is to monitor the health of the structure and assess possible damage by employing a number of sensing devices deemed appropriate for the task. Fig. 1 shows the Surface Inspection Laboratory which emulates the task environment where the inspection will be performed. This figure shows a one-third scale mock-up of part of the truss structure of SSF, as well as a seven degree-of-freedom (DOF) Robotics Research Corporation (RRC) manipulator that carries the inspection sensors at its end-effector. Complete inspection of the truss structure and the orbital replacement units mounted on it requires the capability to reach safely into the truss openings and to maneuver in constricted workspaces inside the truss. Therefore, the ability to perform automated collision-free motions in a congested environment is a fundamental requirement for the project.

The robotic arm used in the project is kinematically redundant since it possesses more DOF than are required to achieve the desired position and orientation of the end-effector. One of the advantages of robot redundancy is the potential to use the "extra" DOF to maneuver in a congested workspace and avoid collisions with obstacles. Much of the work reported to date concerning collision avoidance for manipulators has dealt with high-level *path planning*, in which the end-effector path is planned *off-line* so as to avoid collision with workspace obstacles (see, for example, the recent text [1] and the references therein). Alternatively, the obstacle avoidance problem can be solved *on-line* by the robot controller at the low level. Previous approaches to on-line obstacle avoidance for redundant manipulators have focused on the inverse kinematics portion of the robot control problem (e.g., [2]-[7]), although Khatib [8] and Colbaugh *et al.*

Manuscript received August 20, 1993; revised July 5, 1994.

K. Glass and R. Colbaugh are with the Department of Mechanical Engineering, New Mexico State University, Las Cruces, NM 88003 USA.

D. Lim and H. Seraji are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA.

2 - 11 IEEE Log Number 9410112.

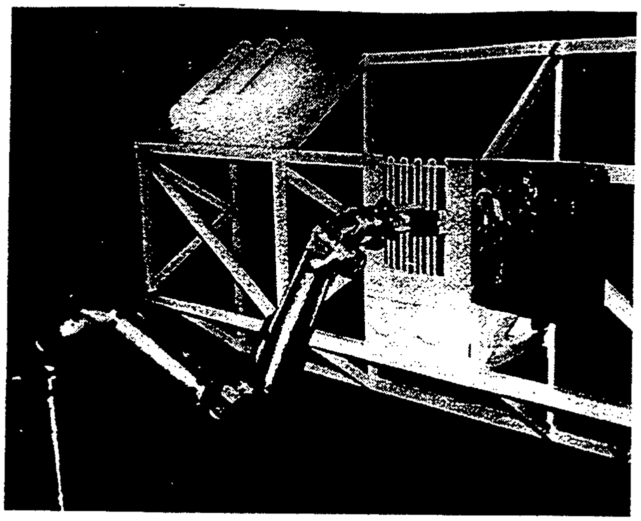


Fig. 1. Surface inspection laboratory at JPL.

[9] have proposed complete control systems for redundant robots maneuvering in the presence of obstacles.

In this paper, attention is focused on the problem of controlling a redundant robot in real-time so that it closely tracks the desired end-effector trajectory and simultaneously avoids workspace obstacles. The proposed approach is to represent the collision avoidance requirement as a set of kinematic inequality constraints, and then ensure that these constraints are satisfied during the desired end-effector motion. The scheme is implemented at the differential inverse kinematic level using a damped-least-squares implementation of *configuration control* [10], [11]; thus it is assumed that a joint-space position controller is available. The simplicity and computational efficiency of the proposed approach allows obstacle avoidance to be implemented in a dynamically varying environment in real-time. The efficacy of the scheme is illustrated through a comprehensive computer simulation and experimental study.[1]

The paper is organized as follows. Section II presents the collision avoidance scheme and provides a brief summary of various implementation issues. The collision avoidance strategy is illustrated in Section III through a computer simulation study and in Section IV through experiments with a Robotics Research 7 DOF arm. Finally, Section V summarizes the paper and draws some conclusions.

II. Collision Avoidance Strategy

Consider the problem of maneuvering a kinematically redundant manipulator in a congested workspace in such a way that the desired end-effector trajectory is closely tracked and, at the same time, collisions between the manipulator links and workspace obstacles are avoided. It is assumed that the robot has low-level joint servo-loops capable of ensuring that the robot joint-space position $\theta \in \Re^n$ closely tracks any desired trajectory $\theta_d(t)$. Thus the above problem becomes one of *kinematic control*, and involves specifying $\theta_d(t)$ so that the end-effector trajectory tracking/collision avoidance objectives are accomplished simultaneously.

A. Overview of Control Scheme

Let $\mathbf{x} \in \Re^m$ denote the position/orientation of the manipulator end-effector, and recall that $m < n$ because the robot is kinematically redundant. The forward kinematics and differential kinematics for the manipulator can be written as

$$\mathbf{x} = \mathbf{f}(\theta) \tag{1}$$

$$\dot{\mathbf{x}} = J_e(\theta)\dot{\theta} \tag{2}$$

where $\mathbf{f} : \Re^n \rightarrow \Re^m$, and $J_e \in \Re^{m\times n}$ is the end-effector Jacobian matrix. Our approach to kinematic control is to utilize the *improved configuration control method* described in [11]. Briefly, this method involves augmenting the end-effector task with additional user-specified tasks and then inverting the augmented differential kinematics using a damped-least-squares (DLS) solution. Thus the user defines the kinematic functions

$$\mathbf{y} = \mathbf{g}(\theta) \tag{3}$$

where $\mathbf{g} : \Re^n \rightarrow \Re^r$ and r is the number of kinematic functions to be controlled as the additional task. Augmenting (1) with (3) yields the kinematic model

$$\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{f}(\theta)\\ \mathbf{g}(\theta)\end{bmatrix} = \mathbf{h}(\theta) \tag{4}$$

and the differential kinematic model

$$\dot{\mathbf{z}} = \begin{bmatrix}J_e(\theta)\\ J_c(\theta)\end{bmatrix}\dot{\theta} = J(\theta)\dot{\theta} \tag{5}$$

where $\mathbf{z} \in \Re^{(r+m)}$ is the configuration vector and $J_c \in \Re^{r\times n}$ is the Jacobian matrix associated with the additional task.

The desired behavior of the manipulator can now be specified by defining the desired configuration vector $\mathbf{z}_d(t) = [\mathbf{x}_d^T(t)|\mathbf{y}_d^T(t)]^T$. The DLS approach to computing the desired joint-space trajectory $\theta_d(t)$ corresponding to $\mathbf{z}_d$ is to compute $\dot{\theta}_d$ from [11]

$$\dot{\theta}_d = [J^T W J + \lambda^2 W_v]^{-1} J^T W[\dot{\mathbf{z}}_d + K\mathbf{e}] \tag{6}$$

and then integrate it to obtain θ_d. In (6), $\mathbf{e} = \mathbf{z}_d - \mathbf{z}$ is the tracking error, $K \in \Re^{(r+m)\times(r+m)}$ is a symmetric positive-definite position feedback gain that corrects for linearization error and other unmodeled effects, and the relative importance of close trajectory tracking and low joint velocities is reflected in the choice of the symmetric positive-definite weighting matrices $W \in \Re^{(r+m)\times(r+m)}$ and $W_v \in \Re^{n\times n}$ and the positive scalar constant λ. The DLS formulation of the configuration control scheme gives optimal solutions to (5) that are robust to singularities. This is achieved by optimally reducing the joint velocities to induce minimal errors in the task performance by modifying the task trajectories. This approach to kinematic control is discussed in (e.g., [11], [13]-[17]), and various properties and features of this scheme are presented in these references. A few additional facts concerning DLS solutions that are useful in our subsequent development are summarized here; the derivation of these results is given in the Appendix.

First note that the DLS formulation (6) can be written in the following equivalent form (see Appendix A.1)

$$\dot{\theta}_d = W_v^{-1} J^T [J W_v^{-1} J^T + \lambda^2 W^{-1}]^{-1} (\dot{\mathbf{z}}_d + K\mathbf{e}) \tag{7}$$

The formulation (7) possesses certain advantages over (6) when $(r + m) < n$, including enhanced computational efficiency and increased flexibility concerning the selection of W, W_v, and λ (for example, λ can be set to zero, corresponding to the specification that the desired trajectory is to be tracked exactly). Next observe that the matrices W and W_v can be chosen to "scale" the components of $\mathbf{x}$ and θ and thereby provide a natural way to handle difficulties associated with mixed units (e.g., meters and degrees), and also different emphasis in performance requirements (e.g., high accuracy

[1] Recently another approach to combining the strategies [9], [11] to achieve inverse kinematics-based collision avoidance has been proposed in [12]. This work adopts a sensor-based approach and is implemented and verified through computer simulation.

in position tracking and low accuracy in orientation tracking). For example, letting

$$W = \text{diag}[(\text{max. allowed velocity error})_{\dot{z}_i}^{-2}] = W^{1/2}W^{1/2}$$
$$W_v = \text{diag}[(\text{max. allowed joint rate})_{\dot{\theta}_i}^{-2}] = W_v^{1/2}W_v^{1/2} \quad (8)$$

leads naturally to the definition of the "normalized" joint-space and task-space variables $\dot{\theta}_d^* = W_v^{1/2}\dot{\theta}_d$ and $\dot{z}^* = W^{1/2}(\dot{z}_d + K\mathrm{e})$. Note that these quantities are dimensionless in view of (8), so that mixed units are no longer a problem, and that the definitions also permit convenient incorporation of different emphasis in performance requirements (since, for example, each component of the task-space is now scaled by the maximum permissible error). These definitions, together with $J^* = W^{1/2}JW_v^{-1/2}$, permit (6) and (7) to be written as

$$\dot{\theta}_d^* = (J^{*T}J^* + \lambda^2 I)^{-1}J^{*T}\dot{z}_d^* \quad (9)$$
$$= J^{*T}(J^*J^{*T} + \lambda^2 I)^{-1}\dot{z}_d^* \quad (10)$$

where λ reflects the relative significance of tracking errors and joint velocities. Note that the definitions (8) can also reduce the problem of selecting W, W_v, K to one of selecting the two scalars λ and k (since the weighting on the elements of $\dot{e}$ are included in W, it suffices to use $K = kI$). The task of specifying these parameters can be guided by consideration of the following two bounds, which are derived in Appendix A.2:

$$\| \mathrm{e} \| \leq \frac{\lambda^2 \| \dot{z}_d \|}{k\sigma_{min}^2} \quad (11)$$

$$\| \dot{\theta} \| \leq \frac{1}{2\lambda} \| \dot{z}_d + k\mathrm{e} \| \quad (12)$$

where σ_{min} is the minimum singular value of J. Note that, as expected, $\| \mathrm{e} \|$ can be reduced by increasing k and/or decreasing λ, and that these adjustments tend to increase $\| \dot{\theta} \|$. Thus tracking error requirements and joint rate limits can be balanced through the selection of k and λ. It should be mentioned that both k and λ can be adjusted on-line based on the observed performance of the robot (e.g., [11], [14]). This typically leads to improved performance, and if this strategy is adopted the analysis summarized above can be used to select initial values or bounds for these parameters.

B. Summary of Collision Avoidance Scheme

The real-time collision avoidance strategy described in this paper is implemented within the DLS configuration control framework summarized in Section II-A and consists of two steps. The first step utilizes an efficient recursive algorithm to formulate the requirement that no robot link should contact any workspace obstacle. This requirement can be expressed as a set of task-space inequality constraints of the form

$$\mathrm{g}(\theta) \geq 0 \quad (13)$$

where $\mathrm{g} : \Re^n \rightarrow \Re^p$ concisely quantifies the collision avoidance requirement. Constructing these inequalities for a general manipulator moving in a workspace cluttered with convex obstacles contained in spheres is described in [9]. An efficient method for computing these inequalities for anthropomorphic robots maneuvering through openings is presented in Section II-C.

The second step in the collision avoidance scheme is to ensure satisfaction of (13) while tracking the desired end-effector trajectory. This objective is achieved by slightly modifying the DLS configuration control strategy to include inequality constraints. Observe that the inequality constraint (13) can be incorporated into the DLS configuration control framework by introducing the constraint error e_{s_i} defined as follows: $e_{s_i} = 0$ if $g_i \geq 0$, and $e_{s_i} = -g_i$ if $g_i < 0$. Then the DLS formulation (6) can be written as

$$[J^TWJ + W_v + \sum_{i=1}^{p} w_iJ_i^TJ_i]\dot{\theta}_d = J^TW(\dot{z}_d + K\mathrm{e}) + \sum_{i=1}^{p} w_iJ_i^Te_{s_i} \quad (14)$$

where $\lambda = 1$, $J_i = \partial g_i/\partial\theta \in \Re^{1\times n}$, and the w_i are nonnegative scalar weights whose magnitudes indicate the relative importance of inequality $g_i \geq 0$. Note that in the current collision avoidance scheme $\dot{z}_{d_i}$ is set to 0, so that here we are interested in *regulating* a collision avoidance task, not *tracking* a desired evolution. The implementation (14) is illuminating because it makes clear how the inequalities (13) can be introduced and then removed as the task evolves. In addition, this formulation leads to a modular algorithm. It should be clear, however, that the DLS scheme (7) can be modified in an analogous manner, since adding the inequality constraint simply corresponds to augmenting the Jacobian matrix and the task-space error vector when the constraints are "active" and then removing these terms when the constraints are "inactive." As indicated above, (7) possesses computational and implementation advantages over the more standard result (6). These advantages are typically significant in obstacle avoidance applications, since in these applications it is usually the case that $(m+r+p) < n$ (recall that m is the dimension of the task-space, for example the number of end-effector position/orientation coordinates to be controlled, p is the number of collision avoidance constraints currently active, and r is the number of other additional tasks).

Finally, note that although activation and deactivation of constraints tends to be "damped" by the DLS scheme, so that abrupt changes in $\dot{\theta}_d$ are not observed, this activation and deactivation can be smoothed and the overall performance can be enhanced by defining the weights w_i to be smooth functions of the errors e_{s_i} as follows:

$$w_i(e_{s_i}) = \begin{cases} \frac{w_{max}}{2}[1 - \cos\pi(e_{s_i}/e_{max})] & 0 \leq e_{s_i} \leq e_{max} \\ w_{max} & e_{s_i} > e_{max} \end{cases} \quad (15)$$

with w_{max}, e_{max} selected based on the particular application and overall system geometry (note that e_{max} is not the maximum permissible error, but rather the error magnitude beyond which the weight w_i is set to its maximum value).

C. Geometrical Calculations

In order to realize the objective of real-time collision avoidance using modest computing power, we utilize geometrically simple models for objects in the workspace and for the manipulator itself, and adopt an efficient recursive strategy for computing the necessary geometrical relationships between these object models and the manipulator model. Thus, while we use the full robot kinematic model for control purposes, we shall approximate the RRC arm geometry with three line segments for all collision avoidance calculations; these segments are the "fictitious links" connecting frames $\{1\}, \{3\}, \{5\}$, and $\{T\}$, where $\{T\}$ is the user-defined tool frame (see Fig. 2(a)). Examination of the actual kinematics of the RRC arm indicates that this simple model represents a reasonable compromise between geometrical fidelity and computational simplicity. This model implicitly assumes the existence of sufficient "buffer" regions surrounding all obstacles, to account for the physical dimensions of the robot. Thus the buffer thickness is defined to be the maximum discrepancy between the physical dimension of the robot and the simplified "stick-figure" robot (see Fig. 2(a)), and this buffer thickness is incorporated into the model for all workspace obstacles. In view of the nature of this construction, it is recommended that the task be aborted if the constraint error e_{s_i} ever becomes so large that this buffer region is violated. In what follows, four classes of obstacles

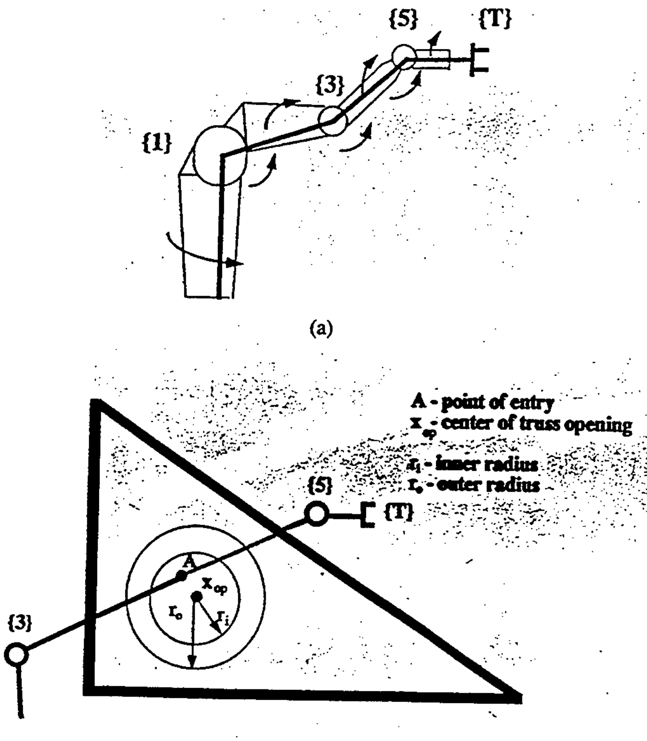


Fig. 2. (a) Robotics Research Corporation K-1207 redundant manipulator approximation. (b) Collision avoidance task description.

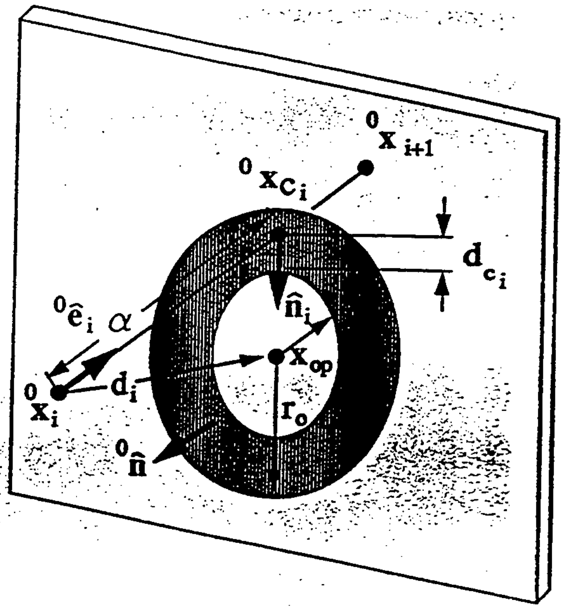


Fig. 3. Collision avoidance parameter descriptions.

are considered: spheres, circular openings, triangular openings, and rectangular openings. Note that by appropriately selecting dimensions and combining these "object primitives," a large class of workspace objects can be modeled with reasonable accuracy.

The results summarized in the previous two sections give a means of simultaneously tracking the desired end-effector trajectory and avoiding arm collisions with workspace obstacles provided that the terms e_{s_i}, J_i $(i = 1, 2, ..., p)$ in (14) can be computed efficiently. Algorithms for the calculation of these quantities for spherical obstacles are given in [9]. The remainder of this section is focused on calculating these terms for the RRC manipulator maneuvering through circular, triangular, and rectangular openings. Consider the schematic of the RRC arm maneuvering through a circular opening in a planar structure shown in Fig. 2(b). We assume that the RRC arm forward kinematics is available in the form of homogeneous transforms ${}^0_iT(\theta)\ \forall i$, where $\{0\}$ is the user-defined world frame. Thus the positions of the origins of frames $\{1\}, \{3\}, \{5\}, \{T\}$ are known, and we denote these positions by ${}^0\mathbf{x}_1, {}^0\mathbf{x}_2, {}^0\mathbf{x}_3, {}^0\mathbf{x}_4 \in \Re^3$.

For clarity of exposition, we idealize the problem of maneuvering through an opening with a robot by considering an infinite planar surface with one opening possessing the shape of a circle, a triangle, or a rectangle. Note that this problem is actually fairly general, and that its solution can readily be extended to other opening geometries and to multiple openings (the assumption of an infinite surface with a single opening is conservative in that, for instance, it does not permit an "elbow" to back into another, fortuitously located, opening). In the case of a circular opening, we assume the geometry is parameterized with the triple ${}^0\mathbf{x}_{op}, \mathbf{r}, {}^0\hat{\mathbf{n}}$, where ${}^0\mathbf{x}_{op} \in \Re^3$ locates the center of the opening, $\mathbf{r} = [r_o\ r_i]^T \in \Re^2$ gives the opening outer and inner radii, and ${}^0\hat{\mathbf{n}} \in \Re^3$ is the unit vector normal to the surface on the $\{0\}$ side of the planar surface. For triangular or rectangular openings, it is assumed that the vertices of the polygon are known. In these cases, the vertices can be readily utilized to compute ${}^0\hat{\mathbf{n}}$ and one vertex can be (arbitrarily) identified as ${}^0\mathbf{x}_{op}$ [18].

Our objective is to ensure that the robot tracks the desired end-effector trajectory while avoiding collisions with either the planar surface or the sides of the opening. We consider the problem of avoiding contact with the planar surface first. A consequence of assuming "large" surfaces is that the set of points with the potential for colliding with the surface is reduced to ${}^0\mathbf{x}_1, {}^0\mathbf{x}_2, {}^0\mathbf{x}_3, {}^0\mathbf{x}_4$. We define

$$d_i = {}^0\hat{\mathbf{n}}^T({}^0\mathbf{x}_i - {}^0\mathbf{x}_{op}) \qquad i = 1, 2, 3, 4 \tag{16}$$

to be the (signed) distances from the planar surface to the ${}^0\mathbf{x}_i$, and note that $d_i > 0$ indicates that ${}^0\mathbf{x}_i$ is on the $\{0\}$ side of the plane; it is assumed that initially $d_i > 0\ \forall i$. Our approach to avoiding collisions with the planar surface is to monitor the d_i as the task evolves, and allow this information to dictate subsequent action. For example, if $d_i < 0$ and $d_{i+1} > 0$ then it is assumed that ${}^0\mathbf{x}_i$ is colliding with the surface and is not attempting to maneuver through the opening. In this case, $e_{s_i} = -d_i$, ${}^0\hat{\mathbf{n}} = {}^0\hat{\mathbf{n}}_i$, and the scheme developed for spherical obstacles is directly relevant here [9]. Alternatively, if $d_i > 0$ and $d_{i+1} < 0$, it is assumed that an attempt is being made to maneuver through the opening, and the strategy developed below for avoiding collisions with the opening is invoked. Implicit in this procedure is the assumption that ${}^0\mathbf{x}_4$ (that is, $\{T\}$) is being directed in a reasonably well-informed manner. Note that in the ideal case in which the above assumptions (zero thickness links, openings in planar surfaces, well-informed end-effector motion) are satisfied, the proposed strategy will always detect collision when it is eminent. In practical cases, this approach permits the development of a simple, real-time, control-level approach to collision avoidance which will work for a large class of obstacles.

Our approach to avoiding collisions between the arm and the sides of the opening is to utilize the d_i computed in (16) to identify a *critical link*. More specifically, if $d_i > 0$ and $d_{i+1} < 0$ for any i, then the link i connecting ${}^0\mathbf{x}_i$ and ${}^0\mathbf{x}_{i+1}$ is a *critical link* and potentially contains a *critical point* (see Fig. 3 for an illustration of a critical point ${}^0\mathbf{x}_c$; a precise definition of this term is given in (17) below, and additional details regarding the nature and role of critical points in collision avoidance are provided in [9]). Thus a link contains a critical point candidate once it has intersected the plane of

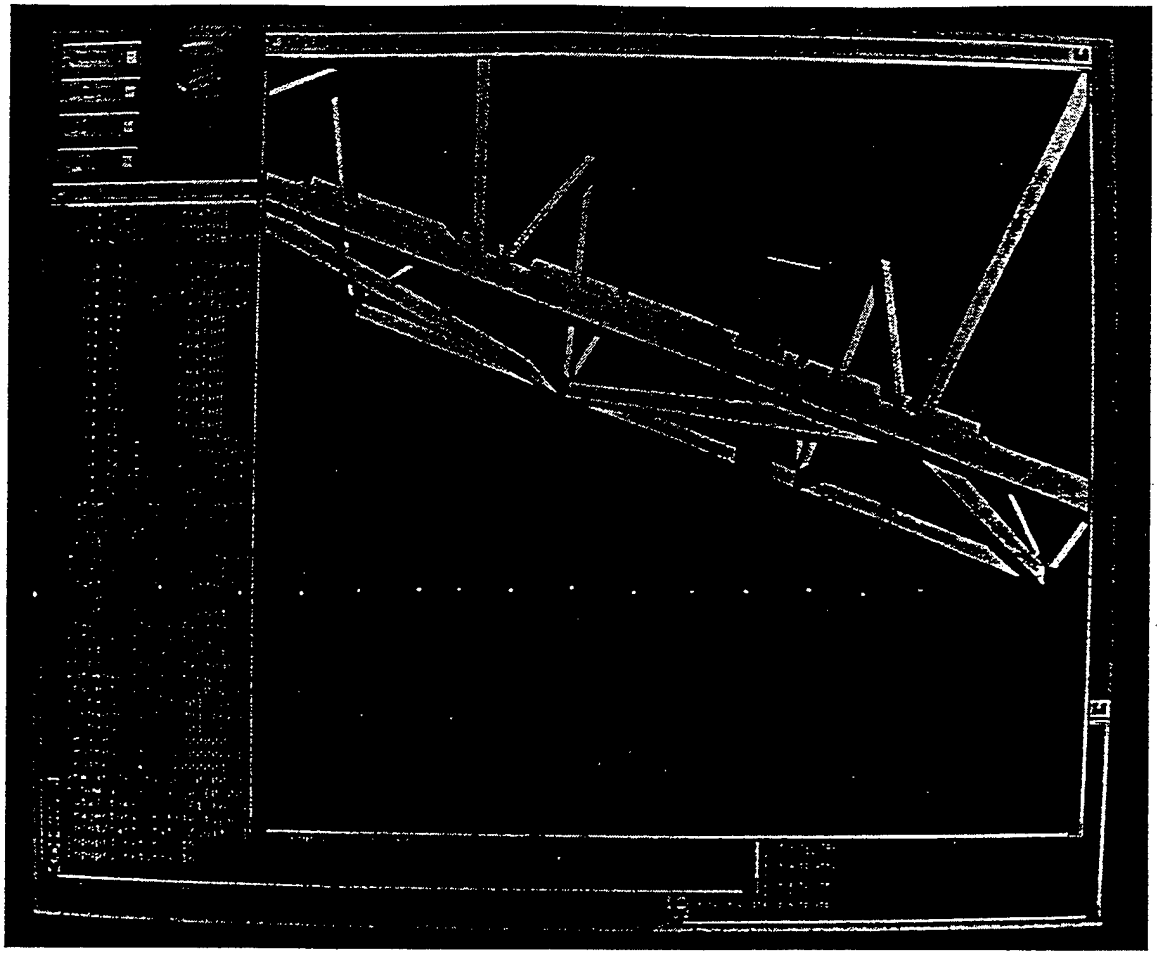


Fig. 4. Graphical depiction of RRC arm and truss structure.

the opening. Given the assumption that the obstacle parameterizations are known, we desire to compute the following quantities for each link: the location of the critical point candidate $x_{c_i} \in \Re^3$ for the link i, the distance $d_{c_i} \in \Re^+$ between x_{c_i} and x_{op}, and the unit vector $\widehat{n}_i \in \Re^3$ pointing from x_{c_i} to x_{op} (see Fig. 3). These quantities can be computed recursively starting from the robot base. Note that if more than one link is critical it is recommended that the task be aborted.

Once a critical link has been identified, it is necessary to determine the terms e_{s_i} and J_i required by the control scheme. Because the nature of these calculations varies with the geometry of the opening, we consider the three types of openings separately.

Circular Opening

Here the location of the critical point candidate is

$$
\begin{aligned}
{}^0x_{c_i} &= {}^0x_i + \alpha\, {}^0\widehat{e}_i \qquad (17)\\
\text{where}\quad {}^0\widehat{e}_i &= ({}^0x_{i+1} - {}^0x_i)/\,\|{}^0x_{i+1} - {}^0x_i\|\\
\alpha &= -d_i / {}^0\widehat{e}_i^T\, {}^0\widehat{n}
\end{aligned}
$$

Then the critical distance is $d_{c_i} = r_i - \|{}^0x_{c_i} - {}^0x_{op}\|$, the critical point candidate is a critical point if $d_{c_i} < 0$, and if the point is critical then $e_{s_i} = -d_{c_i}$. The corresponding row of the Jacobian for this point can then be calculated using the following algorithm:

1) Define $J^{**} \in \Re^{3\times 7}$ as ${}^0\dot{x}_{c_i} = J^{**}\dot{\theta}$ and compute its columns $J_j^{**} = {}^0\widehat{z}_j \times ({}^0x_{c_i} - {}^0x_j)$ where ${}^0\widehat{z}_j$ is the z-axis unit vector for $\{j\}$ and it is understood that $J_j^{**} = 0$ for each $j > i$.
2) Define $\widehat{n}_i = ({}^0x_{op} - {}^0x_{c_i})/\,\|{}^0x_{op} - {}^0x_{c_i}\|$ and compute the jth element of J_i as $J_{ij} = \widehat{n}_i^T J_j^{**}, j = 1, 2, \ldots, 7$.

This algorithm is computationally simple, recursive, and robust, and examples of its implementation are provided in Sections III and IV.

Triangular and Rectangular Openings

Here the critical point candidate can be located using the scheme (17) derived for circular openings, but the determination of e_{s_i} and J_i is slightly more involved because these openings do not possess the symmetry of a circle. Our basic approach to determining these quantities is to specify local coordinate systems at the polygon vertices, using an off-line calculation, and then to represent ${}^0x_{c_i}$ in these new frames; once these coordinate transformations have been completed, $e_{s_i}, \widehat{n}_i$ can be computed easily, and the J_i can be constructed using the algorithm given for circular openings [18]. It should be mentioned that the scheme developed for triangular and rectangular openings actually works for any convex polyhedron.

In all cases where we desire the manipulator to reach through an opening, the collision avoidance scheme outlined above provides an efficient and flexible environment. Flexibility can be achieved by noting that the scheme can be modified for model-based or sensor-based manipulator control. In model-based control where information is available regarding the environment, the collision avoidance scheme described above allows the robot maximal use of its workspace, since obstacles are treated as inequality constraints. For example, the collision avoidance scheme for a triangular opening utilizes knowledge of the location of each of the three vertices of the triangle, and in this case the manipulator has the benefit of a large workspace at the expense of requiring extensive information regarding the world. If less information is available, the triangular

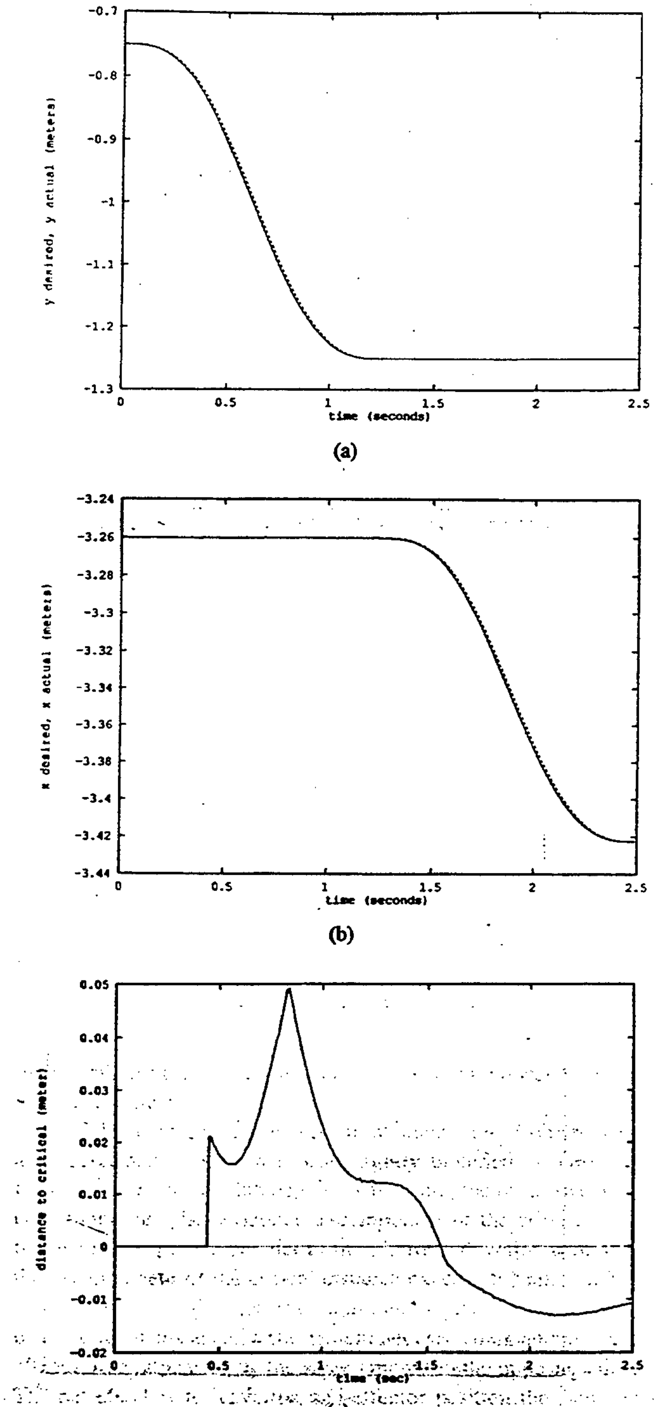


Fig. 5. (a) Response of end-effector coordinate y in simulation example; (b) response of end-effector coordinate x in simulation example; (c) critical point distance d_{c_i} in simulation example

opening can be modeled as a point in the plane. Then the collision avoidance scheme for a circular opening can be used with a zero inner radius and a small outer radius. This has the effect of defining a region that the manipulator critical point should *pass-through*. Here much less information is needed regarding the boundary of the opening, but the effective workspace of the robot is decreased.

III. Simulation Results

The robot collision avoidance strategy described in Section II is now applied to the RRC manipulator (7DOF) mounted on a mobile platform (1DOF) in a computer simulation study. The simulation environment consists of a Personal Iris Graphics Workstation, the *Hydra* graphics rendering package developed at JPL, and the software necessary to realize the DLS configuration control implementation of the collision avoidance strategy.

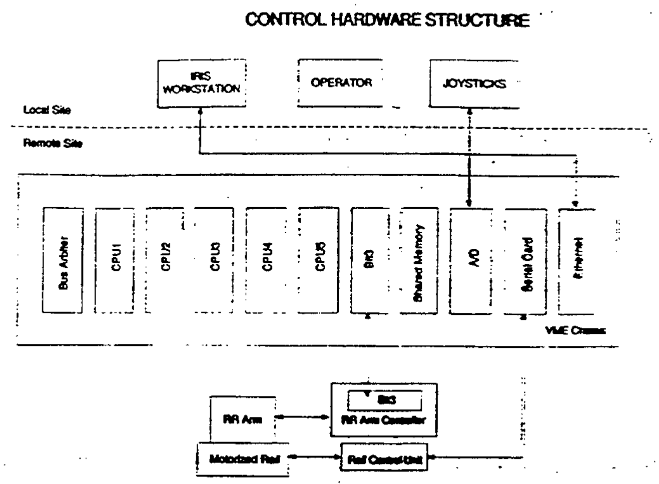


Fig. 6. Control hardware diagram.

For simulation of control algorithms and for previewing motions of the manipulator-plus-platform system, a graphical simulation capability has been developed on the IRIS Workstation. The Hydra graphics rendering package allows the user to: (1) build graphical models of typical robot components and workspace structures, (2) input relative motions to the components, and (3) change the viewing perspective of the workspace of the robot. Thus, the combination of the DLS control implementation of the collision avoidance scheme and the Hydra graphics rendering capabilities allow for complete robot system simulation.

The objective of the simulation example is to illustrate the collision avoidance capability of the RRC manipulator and mobile platform for the task of reaching inside the truss structure depicted graphically in Fig. 4. This truss structure is one of the components in the RSI Laboratory at JPL, and in this example we desire to enter the interior of the truss and then move around in the interior to simulate an internal inspection task. Thus this task requires the robot to track a desired position trajectory while simultaneously avoiding collision with the beams of the truss openings.

In this simulation, the DLS configuration control approach to obstacle avoidance summarized in Section II-B is implemented with the 8 DOF system. Here the triangular opening *pass-through* method is employed to model the situation in which reduced information is available regarding the truss structure geometry. The triangular opening in the truss has vertices ${}^{0}\mathbf{p}_1 = [-3.08\ -0.86\ 1.05]^T, {}^{0}\mathbf{p}_2 = [-3.08\ -0.86\ 0.55]^T, {}^{0}\mathbf{p}_3 = [-3.96\ -0.86\ 0.55]^T$ (relative to the user-specified world frame and with meter the unit of length) and through simple calculation the center point equidistant from the sides of the triangle is found to be ${}^{0}\mathbf{x}_{op} = [-3.29\ -0.86\ 0.77]^T$. The outer and inner radii are defined as $\mathbf{r} = [0.09\ 0.05]^T$, e_{max} is set as $e_{max} = 0.04$m, and the obstacle surface normal is ${}^{0}\hat{\mathbf{n}} = [0\ 1\ 0]^T$. The inner radius is relaxed (from $r_i = 0.0$ to $r_i = 0.05$) to allow the robot to make larger use of the workspace to complete the task: once a link is critical it is allowed to deviate five centimeters from the "pass-through" point before the obstacle avoidance task is activated. Given the modeling assumptions and the buffer thickness, in the worst case the upper arm of the manipulator would be permitted to approach the truss structure to within 5 centimeters before the task is aborted. In this simulation, from an initial end-effector position of $\mathbf{x} = [-3.29\ -0.80\ 0.77]^T$ we want the robot to move

0.50m in the y-direction and then -0.162m in the x-direction and -0.118m in the z-direction; thus the arm will reach 0.45 meters inside the truss and then move 0.20 meters parallel to the diagonal element of the truss. The desired end-effector trajectory is given as $\mathbf{x}_d(t) = [0.0\ -0.50(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0]^T$ for $t \subset [0, 1.25]$, $\tau = 1.25$, and $\mathbf{x}_d(t) = [-0.162(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0\ -0.118(t/\tau - sin(2\pi t/\tau)/2\pi)]^T$ for $t \subset [1.25, 2.50]$, $\tau = 1.25$. To test the Cartesian positioning capability of the arm/platform system, the orientation control is turned off for this simulation. Thus, the Cartesian task dimension is defined as $m = 3$. When the collision avoidance inequality constraint is activated, $r = 1$ and the task-space DOF are increased to $m + r = 4$. The weights for the DLS configuration control strategy (14) are chosen as $K = 100$, $W = 100$, $W_v = 1$, and $w_{max} = 1000$. Figs. 5(a) and 5(b) show the evolution of the trajectory in the y and x directions, respectively, and Fig. 5(c) shows the critical point distance d_{c_i} for the collision avoidance subtask. From these figures, it is seen that the end-effector task and the collision avoidance requirement are satisfied simultaneously with very little error in the end-effector trajectory tracking.

IV. Experimental Results

In this section, we describe the basic features of the manipulator control system in the JPL RSI Laboratory and summarize an experimental study using the collision avoidance scheme. The manipulator system is shown in Fig. 1 and a control hardware diagram is shown in Fig. 6. The experimental environment consists of an RRC Model K-1207 (7DOF) arm/control unit, a VME chassis with five Heurikon MC68020 processor boards and additional interface cards, two joysticks, a motorized platform/control unit on which the arm is mounted, and a Silicon Graphics IRIS Workstation. The truss structure in the Laboratory has eight triangular openings, and in this experiment we want the robot to enter the truss and track a trajectory inside the truss structure.

As shown in Fig. 2(a), the dexterous manipulator has seven revolute joints in an alternating roll/pitch sequence beginning with the shoulder roll at the base and ending with the tool-plate roll at the hand. The arm pedestal is mounted on a motorized mobile platform which provides one additional translational degree-of-freedom that can be treated as a prismatic joint. The complete manipulator system therefore has eight independent joint degrees-of-freedom; the system has two degrees-of-redundancy since six joints are sufficient for the basic task of end-effector position and orientation in the three-dimensional workspace. Although the real-time system is capable of moving the platform, unlike in the simulation environment the platform is not treated as an 8th degree-of-freedom since the present serial interface to the platform is too slow for the platform to be considered as a true joint.

The Robotics Research arm is controlled by a real-time microprocessor-based controller that interfaces directly with the Multibus-based arm control unit supplied by the manufacturer (see Fig. 6). The real-time controller is a VMEbus-based system that uses five Motorola MC68020 processors along with various data acquisition, memory, and communication devices. The VME controller is linked via socket communication to the Silicon Graphics IRIS Workstation, which serves as the host computer for the graphical user interface. All of the control software is written in the C language and executes on the VME controller running the VxWorks real-time operating system. The real-time VME chassis communicates with the arm control unit (Multibus chassis) via a two-card VME-to-Multibus adaptor from the Bit 3 Corporation. This enables joint set-points to be sent to the arm control unit at a high speed (400 Hz) via a shared memory servo-level interface. Thus, the arm control unit is a low-level joint-space position controller with the higher level control being done on the VME chassis. The high-level control loop, which consists of a trajectory generator, forward kinematics, and DLS configuration control algorithm, runs on a single Heurikon 68020 board at 44 Hertz (22.7 milliseconds). The portion of this time used to calculate the collision avoidance task is 3.5 milliseconds. The other four processors are used for servicing hardware devices. These hardware devices are an Ethernet board (for user input), an analog input board (for the joysticks), a serial board (for the

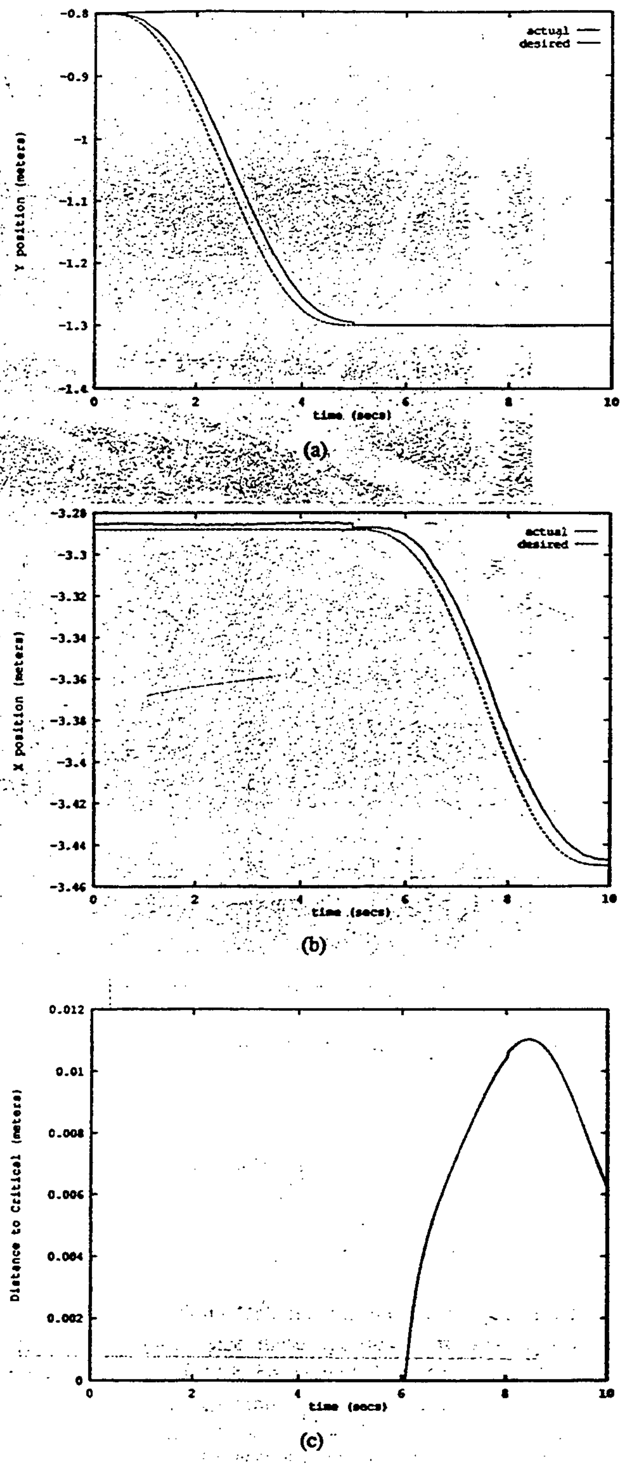
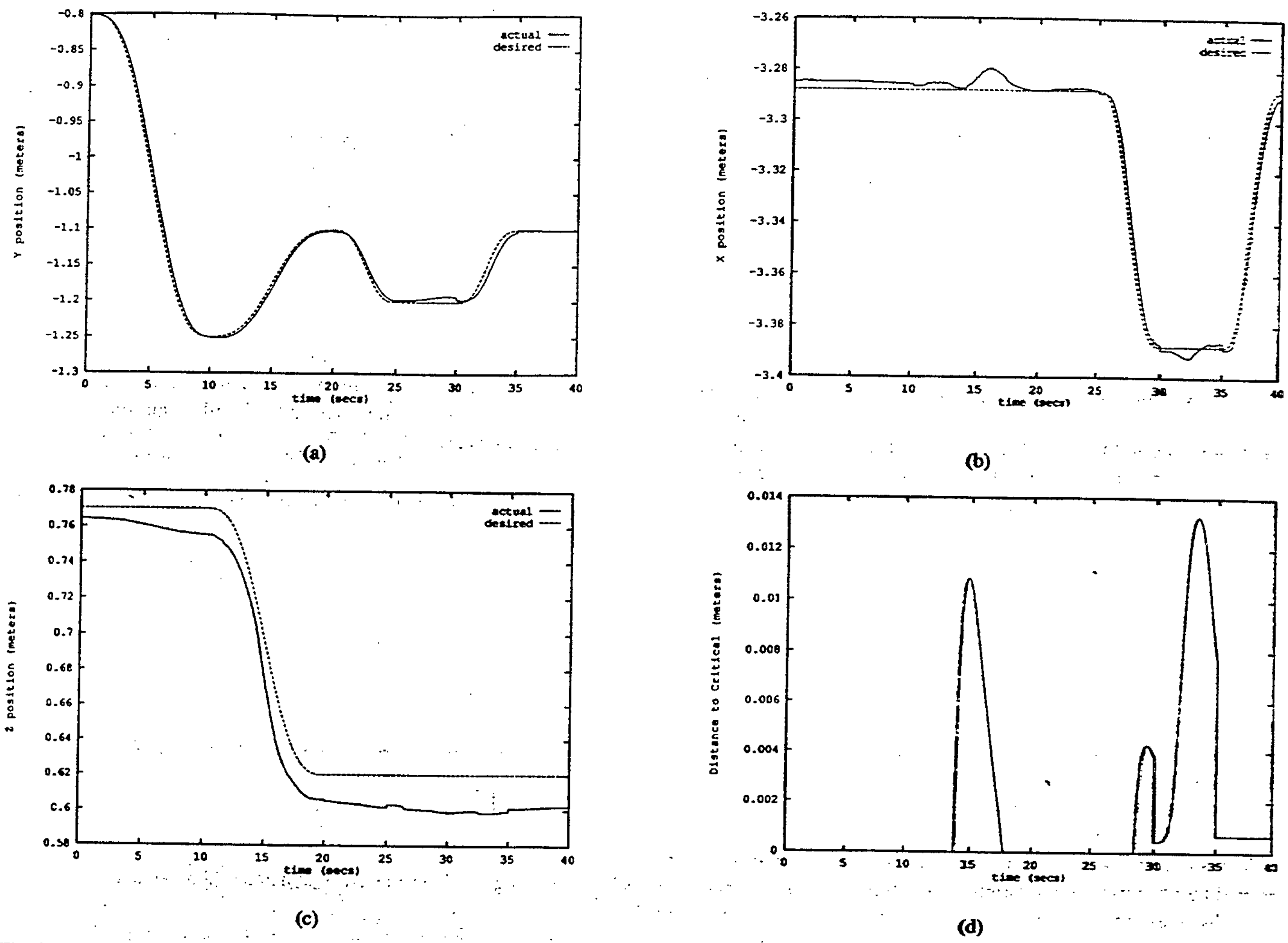


Fig. 7. (a) Response of end-effector coordinate y in experiment one; (b) response of end-effector coordinate x in experiment one; (c) critical point distance d_{c_i} in experiment one.



(a)

(b)

(c)

(d)

Fig. 8. (a) Response of end-effector coordinate y in experiment two; (b) response of end-effector coordinate x in experiment two; (c) response of end-effector coordinate z in experiment two; critical point distance d_{c_i} in experiment two.

mobile platform), and a Bit3 board (for communication with the RRC controller).

The DLS configuration control implementation of obstacle avoidance described in Section II was slightly modified in these experiments. To simulate the 8th degree-of-freedom, the obstacle avoidance code monitored the horizontal x-component of the critical distance; this is the component parallel to the platform direction of motion. If the x-component of the critical distance exceeds 0.10m, a command is issued to move the platform an amount equal to the x-component of the critical distance, while simultaneously commanding the end-effector to compensate by the same amount in the opposite direction. The net effect is to leave the end-effector position the same, while attempting to lessen the horizontal component of the critical distance. This initial implementation provides a simple heuristic model for simulating the 8 degrees-of-freedom with the current experimental setup. In the experimental study, many different tasks were studied. Here two examples are chosen for presentation to illustrate the flexibility of the collision avoidance scheme and the DLS implementation of configuration control. In each example, the collision avoidance strategy is implemented using the triangular opening *pass-through* model of the obstacle and the obstacle geometry and the initial configuration of the arm and platform are exactly the same as in the simulation study.

In the first experiment, from an initial end-effector position of $\mathbf{x} = [-3.29\ -0.80\ 0.77]^T$ the end-effector is commanded to move -0.50m in the y-direction and then −0.162m in the x-direction and −0.118m in the z-direction, and orientation control of the end-effector is not activated. This motion will cause the arm to reach 0.45 meters inside the truss and then move parallel to the diagonal of the triangular opening. Thus, the trajectory is specified to produce the same motion as in the simulation example only at a slower rate. The desired end-effector trajectory is specified as $\mathbf{x}_d(t) = [0.0\ -0.50(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0]^T$ for $t \subset [0, 5.0]$, $\tau = 5.0$, and $\mathbf{x}_d(t) = [-0.162(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0\ -0.118(t/\tau - sin(2\pi t/\tau)/2\pi)]^T$ for $t \subset [5.0, 10.0]$, $\tau = 5.0$. The weights for the DLS configuration control strategy (14),(15) are chosen as $K = 1.0$, $W = 1.0, W_v = 0.005$, and $w_{max} = 10$. Figs. 7(a) and 7(b) show the evolution of the trajectory in the y and x-directions, respectively, and Fig. 7(c) shows the critical point distance d_{c_i} for the collision avoidance subtask. These figures show that trajectory tracking is reasonable, with the end-effector tracking the desired trajectory to within 0.01m in the x and y-directions, and 0.02m in the z-direction. The obstacle avoidance algorithm maintains the critical distance within a band of 0.012m (the critical distance is zero if the critical point is within the inner radius of 0.05m).

In the second experiment, an inspection-type task is executed. The first two moves will preposition the arm for the internal inspection. On the first move, the arm enters the truss under position and orientation control. For the second move, the wrist joint is pitched down until the xy-plane of the tool frame is parallel to the xy-plane of the world frame. The internal inspection task comprises the next four moves. Here the end-effector task is to trace out a square under position and orientation control, thereby simulating the inspection of a flat surface inside the truss. The initial end-effector position is identical to the first experiment. The first move inserts the arm into the truss and is defined by $\mathbf{x}_d(t) = [0.0\ -0.45(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0]^T$ for $t \subset [0, 10.0]$, $\tau = 10.0$. The second trajectory pitches the wrist down while moving the y and z-coordinates, according to $\mathbf{x}_d(t) = [0.0\ 0.15(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.15(t/\tau - sin(2\pi t/\tau)/2\pi)]^T$ for $t \subset [10.0, 20.0]$, $\tau = 10.0$. The inspection move trajectories trace out a square: $\mathbf{x}_d(t) = [0.0\ -0.10(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0]^T$ for $t \subset [20.0, 25.0]$, $\tau = 5.0$, $\mathbf{x}_d(t) = [-0.10(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0\ 0.0]^T$ for $t \subset [25.0, 30.0]$, $\tau = 5.0$, $\mathbf{x}_d(t) = [0.0\ 0.10(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0]^T$ for $t \subset [30.0, 35.0]$, $\tau = 5.0$, and $\mathbf{x}_d(t) = [0.10(t/\tau - sin(2\pi t/\tau)/2\pi)\ 0.0\ 0.0]^T$ for $t \subset [35.0, 40.0]$, $\tau = 5.0$. Fig. 8(a), 8(b), and 8(c) show the evolution of the end-effector trajectory in the y, x, and z-directions, respectively, and Fig. 8(d) shows the critical point distance d_{c_i} for the collision avoidance subtask. Again the figures show that the obstacle avoidance task is achieved by keeping the critical distance within 0.014m. End-effector trajectory tracking is good for the majority of the trajectory except when $t = 15$ secs and again at $t = 32$ secs, where tracking errors in x is observed. At these times, the critical distance plot shows that the DLS algorithm optimally sacrifices end-effector trajectory tracking to ensure satisfaction of the collision avoidance constraint.

V. Conclusions

This paper presents an efficient, flexible, and robust scheme for real-time collision avoidance for redundant manipulators at the joint-control level. The control scheme is derived at the inverse kinematics level using a DLS implementation of configuration control, and is intended for implementation with a high-level path planning system or with a teleoperation system. The collision avoidance strategy can be modified for use with a variety of obstacle geometries, including solid obstacles and openings. The configuration control framework also allows the user to trade-off end-effector tracking with collision avoidance in cases when both tasks cannot be achieved accurately. Additionally, the scheme provides the user with the capability to vary the level of information needed by the algorithm, so that the approach is equally applicable to model-based control or sensor-based control. The algorithm presented here has been tested extensively in simulations as well as in experiments, and the results of these studies confirm the efficacy of the method.

Acknowledgment

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Additional support was provided by the Army Research Office and a NASA/ASEE Summer Faculty Fellowship held by the first author.

Appendix

Some Properties of the DLS Configuration Control Scheme

In this Appendix, some properties of the DLS configuration control approach to kinematic control are briefly summarized.

A.1 We first derive the equivalent formulation (7) of the DLS scheme (6), and begin by noting that (6) is the solution to the following optimization problem:

$$\text{minimize} \quad \{\lambda^2 \dot{\theta}_d^T W_v \dot{\theta}_d + (\dot{z}_d + K\mathbf{e} - J\dot{\theta}_d)^T W(\dot{z}_d + K\mathbf{e} - J\dot{\theta}_d)\} \tag{A1}$$

Observe that the factorization $W = W_x^T W_x$ and $W_v = W_\theta^T W_\theta$ are possible since $W_v = W_v^T > 0$ and $W = W^T > 0$. Then the definitions $\dot{\phi} = W_\theta \dot{\theta}_d$, $\dot{\mathbf{u}}_d^* = W_x(\dot{z}_d + K\mathbf{e})$, and $J^* = W_x J W_\theta^{-1}$ permit (A1) to be rewritten as

$$\text{minimize}\{\lambda^2 \| \dot{\phi} \|^2 + \| \dot{\mathbf{u}}_d^* \|^2 + \| J^* \dot{\phi} \|^2 - 2\dot{\mathbf{u}}_d^{*T} J^* \dot{\phi}\} \tag{A2}$$

which admits the solution

$$\dot{\phi} = (J^{*T} J^* + \lambda^2 I)^{-1} J^{*T} \dot{\mathbf{u}}_d^* \tag{A3}$$

$$= J^{*T}(J^* J^{*T} + \lambda^2 I)^{-1} \dot{\mathbf{u}}_d^* \tag{A4}$$

where the equivalence of (A3) and (A4) was first noted in [14]. The substitution of the relations $\dot{\phi} = W_\theta \dot{\theta}_d$, $\dot{\mathbf{u}}_d^* = W_x(\dot{z}_d + K\mathbf{e})$, $J^* = W_x J W_\theta^{-1}$ into (A4) then yields (7).

A.2 We now derive the bounds (11),(12) for the DLS schemes (9),(10). We proceed by examining the convergence properties of the DLS scheme (9), rewritten here in a slightly simplified form for convenience of analysis:

$$\dot{\theta} = [J^T J + \lambda^2 I]^{-1} J^T (\dot{z}_d + k\mathbf{e}) \tag{A5}$$

The error dynamic equation that results from applying the control input $\dot{\theta}_d$ calculated using (A5) to the system (5) is

$$\dot{\mathbf{e}} = -kJ(J^T J + \lambda^2 I)^{-1} J^T \mathbf{e} + (I - J(J^T J + \lambda^2 I)^{-1} J^T)\dot{z}_d \tag{A6}$$

where it is assumed that a good position controller is available so that $\theta = \theta_d$. The bound (11) is derived by using a Lyapunov analysis, and toward this end we consider the following Lyapunov function candidate

$$V = \mathbf{e}^T \mathbf{e}/2.$$

Differentiating V along the error dynamics (A6) yields

$$\dot{V} = -k\mathbf{e}^T [J(J^T J + \lambda^2 I)^{-1} J^T]\mathbf{e} + \mathbf{e}^T [I - J(J^T J + \lambda^2 I)^{-1} J^T]\dot{z}_d$$
$$\leq -k\frac{\sigma_{min}^2}{\sigma_{min}^2 + \lambda^2} \| \mathbf{e} \|^2 + \frac{\lambda^2}{\sigma_{min}^2 + \lambda^2} \| \dot{z}_d \| \| \mathbf{e} \| \tag{A7}$$

where σ_{min} is the minimum singular value of J. The upper bound on $\dot{V}$ given in (A7) implies that e is bounded, and permits the following upper-bound on $\| \mathbf{e} \|$ to be established:

$$\| \mathbf{e} \| \leq \frac{\lambda^2 \| \dot{z}_d \|}{k\sigma_{min}^2}$$

To obtain the bound (12), we note from (A5) that

$$\| \dot{\theta} \| = \| (J^T J + \lambda^2 I)^{-1} J^T (\dot{z}_d + k\mathbf{e}) \|$$
$$= \| \sum (\frac{\sigma_i}{\sigma_i^2 + \lambda^2}) \mathbf{v}_i \mathbf{u}_i^T (\dot{z}_d + k\mathbf{e}) \| \tag{A8}$$

where $J = U\Sigma V^T$ with each $\mathbf{u}_i, \mathbf{v}_i$ defined by $U = [\mathbf{u}_1 \mathbf{u}_2 \cdots \mathbf{u}_{m+r}]$ and $V = [\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_n]$. From (A8), it can be seen that the maximum $\| \dot{\theta} \|$ will occur when $(\dot{z}_d + k\mathbf{e})$ aligns with one of the $\mathbf{u}_i$ and $\sigma_i = \lambda$, which leads to

$$\| \dot{\theta} \| \leq \frac{1}{2\lambda} \| \dot{z}_d + k\mathbf{e} \|.$$

Acknowledgment

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Additional support was provided by the Army Research Office and a NASA/ASEE Summer Faculty Fellowship held by the first author.

References

[1] J. Latombe, *Robot Motion Planning.* Boston, MA: Kluwer Academic, Boston, 1990

[2] A. Maciejewski and C. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109-117, 1985,

[3] J. Baillieul, "Avoiding obstacles and resolving kinematic redundancy," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1986, pp. 1698-1704

[4] M. Kircanski and M. Vukobratovic, "Contribution to control of redundant robotic manipulators in an environment with obstacles," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 112-119, 1986.

[5] I. Walker and S. Marcus, "Subtask performance for redundancy resolution for redundant robot manipulators," *IEEE J. Robot. and Automat.*, vol. 4, no. 3, pp. 350-354, 1988.

[6] Z. Guo and T. Hsia, "Joint trajectory generation for redundant robots in an environment with obstacles," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1990, pp. 157-162

[7] H. Zghal, R. Dubey, and J. Euler, "Collision avoidance of a multiple degree of redundancy manipulator operating through a window," *ASME J. Dynam. Syst., Measure., and Cont.*, vol. 114, no. 4, pp. 717-721, 1992.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90-98, 1986.

[9] R. Colbaugh, H. Seraji, and K. Glass, "Obstacle avoidance for redundant robots using configuration control," *J. Robot. Syst.*, vol. 6, no. 6, pp. 721-744, 1989.

[10] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Trans. Robot. and Automat.*, vol. 5, no. 4, pp. 472-490, 1989.

[11] H. Seraji and R. Colbaugh, "Improved configuration control for redundant robots," *J. Robot. Syst.*, vol. 7, no. 6, 1990, pp. 897-928

[12] C. Boddy and J. Taylor, "Whole-arm reactive collision avoidance control of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robot. and Automat.* 1993, pp. (3)382-387

[13] C. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Syst., Man, and Cybern.*, vol. SMC-16, no. 1, pp. 93-101, 1986.

[14] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME J. Dynam. Syst., Measure., and Cont.*, vol. 108, no. 3, pp. 163-171, 1986.

[15] A. Maciejewski and C. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. Robot. Syst.*, vol. 5, no. 6, pp. 527-552, 1988.

[16] O. Egeland, J. Sagli, I. Spangelo, and S. Chiaverini, "A damped least squares solution to redundancy resolution," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1990, pp. 644-649

[17] A. Deo and I. Walker, "Robot subtask performance with singularity robustness using optimal damped least squares," in *Proc. IEEE Int. Conf. Robot. and Automat.* 1992, pp. 434-441.

[18] K. Glass, R. Colbaugh, D. Lim, and H. Seraji, "On-line collision avoidance for redundant manipulators," in *Proc. IEEE Int. Conf. Robot. and Automat.*, 1993, pp. (1)36-43.

A New Adaptive Control Algorithm for Robot Manipulators in Task Space

Gang Feng, *Member, IEEE*

Abstract— Adaptive control of robotic manipulators in task space coordinates is considered in this paper. A new composite adaptive control law, which uses the prediction error and tracking error to drive parameter estimation, is developed based on sliding mode and a general Lyapunov-like concept. It is shown that global stability and convergence can be achieved for the adaptive control algorithm. The algorithm has the advantage that inverse of Jacobian matrix and the bounded inverse of the estimated inertia matrix are not required. The algorithm is further modified so as to achieve robustness to bounded disturbances. A simulation example is provided to demonstrate the performance of the proposed algorithm.

I. Introduction

Adaptive control of rigid robot manipulators has been the interests of many researchers for several years. A recent survey [1] indicates the breadth of the approaches taken. In joint space, one of the early applications of adaptive control to the manipulator problem came from Horowitz and Tomizuka [2] in 1980. Their scheme assumed the plant inertia to be time-invariant and assumed the gravity/friction torques to be known. Later many other researchers have considered controlling a linear perturbational/variational model, and controlling the full nonlinear manipulator with adaptive "high gain," see [3]–[8]. Recently several globally convergent adaptive control results for manipulators appeared [9]–[13], which were summarized in [14]. One of the results is from Craig *et al.* [9], which relies on the two restrictive assumptions. The first is that the joint acceleration is measurable and the second is that the inverse of the estimated inertia matrix remains bounded. The need to measure joint acceleration can be overcome by the introduction of a first order filter as in [10]. The second assumption is more restrictive and requires modification of the parameter update law, and it can be overcome with an algorithm introduced by Spong and Ortega [15]. Their idea is to use the fixed estimate instead of varying estimate for inertia matrix and therefore overcome the drawback of requirement of boundedness of the estimated inertia matrix. Slotine and Li [11] proposed an adaptive control algorithm which requires neither the measurements of the joint accelerations nor the bounded inverse of the estimated inertia matrix.

A composite adaptive control law in joint space was proposed in [16], which used the prediction error and the joint space tracking error to drive the parameter estimator. This design is based on the observation that the parameter uncertainty is reflected in both the tracking error and the prediction error. Therefore it is desirable and reasonable to extract the parameter information from both of sources. In addition, Reed and Ioannou [17] have discussed the robustness issues of adaptive control algorithms for robotic manipulators. They applied σ-modification robust adaptive control strategy to achieve this objective.

However in practice, robot motion is basically defined by the motion of its end effector, i.e. in robot control, the major concern is that the end-effector motion tracks its desired motion, defined in the task space or the so-called Cartesian space.

Manuscript received November 2, 1992; revised March 1, 1994.

The author is with the Dept. of Systems and Control, School of Electrical Engineering, University of New South Wales, Sydney, NSW 2052, Australia.

IEEE Log Number 9409240.

submitted for publication

ADAPTIVE STABILIZATION OF MECHANICAL SYSTEMS USING ONLY CONFIGURATION MEASUREMENTS

R. Colbaugh K. Glass

Department of Mechanical Engineering
New Mexico State University, Las Cruces, NM 88003 USA

Abstract

This paper presents a new approach to stabilizing uncertain mechanical systems using only measurements of the system configuration. The proposed controller is computationally simple, does not require knowledge of the system dynamic model, and can be implemented with a wide variety of systems. It is shown that the control strategy guarantees semiglobal uniform boundedness of all signals and convergence of the system configuration error to zero. If bounded external disturbances are present, a slight modification to the control scheme ensures that uniform boundedness of all signals is retained and that arbitrarily accurate stabilization of the configuration error is achieved. The capabilities of the proposed controller are illustrated though both computer simulations and experiments involving applications of importance in manufacturing. These studies demonstrate that the controller provides a simple and effective means of obtaining high performance error regulation; additionally, the case studies indicate that accurate trajectory tracking can also be realized with the proposed scheme.

1. Introduction

The promise of improved productivity, enhanced reliability, reduced costs, and increased human safety has motivated the introduction of computer-controlled mechanical systems in the manufacturing sector, in space, and in applications involving hazardous environments. This implementation of mechanical systems has revealed that, in many applications, achieving the anticipated benefits requires the development of control systems capable of providing good performance in the presence of incomplete information regarding the system and its environment. Additionally, this work has indicated the importance of the fundamental task of *system stabilization*, that is, of devising control laws which ensure that any prescribed system state becomes an asymptotically stable equilibrium for the controlled system.

This paper considers the problem of stabilizing mechanical systems which have at least one actuator for each configuration degree-of-freedom. These systems are typically referred to as *fully actuated* mechanical systems, and are defined more precisely in the following section. For now it is simply noted that the class is quite large and includes virtually all systems of importance in manufacturing, such as robotic manipulators and machine tools. Most research on the stabilization of mechanical systems has focused on those systems for which complete model information and state measurements are available; work in this area includes investigations of general classes of mechanical systems [e.g., 1-6] as well as studies focusing on particular systems, such as robotic manipulators [7-9] and spacecraft [10,11]. As a result of these studies, it is now well-known that simple proportional-derivative (PD) feedback controllers are capable of globally stabilizing fully actuated mechanical systems, provided that the effects of any system potential energy (such as gravity) are compensated. Ordinarily the requisite potential energy compensation is achieved by including a model of the (gradient of the) potential energy directly in the control scheme, so that the resulting controller possesses a PD-plus-potential-model structure.

While the PD-plus-potential-model stabilization strategy is simple, elegant, and intuitively appealing, there are two potential difficulties associated with this approach. First, these controllers include a linear state feedback component and therefore require full state measurement, which can be problematic in practice. For example, although robotic manipulators and machine tools are generally equipped with high-precision sensors capable of accurately measuring system position, determining the system velocity is more challenging and typically the velocity measurements are contaminated with noise [12]. In fact, velocity sensors are frequently omitted altogether in the interest of saving money, volume, and weight in construction [12]. The second drawback with PD-plus-potential-model stabilization schemes is the need to include the system potential energy model in the control law. Observe that this approach to compensating the effects of the system potential energy can be undesirable because its implementation requires precise knowledge of the structure and parameter values of the potential energy model. This is particularly restrictive for manufacturing automation applications, because in typical tasks the manufacturing system encounters many different tools, payloads, and workpieces, and it is unrealistic to assume that the properties of all of these are accurately known. In addition, this requirement limits the modularity and portability of the controllers, so that implementation of these schemes with a new system can involve considerable effort.

Recognizing the aforementioned difficulties, several researchers have studied methods of addressing one or the other problem. For instance, recently several schemes have been proposed for controlling robotic manipulators using only position measurements [13-17]. One approach to this problem is to design an observer for estimating the velocity utilizing the available (high-quality) position information, and then to use this estimated velocity in place of the actual velocity in the state feedback component of the control law; this strategy is used in [13-15] to solve the manipulator tracking problem. Alternatively, the simple structure of the PD-plus-potential-model solution to the mechanical system stabilization problem suggests a correspondingly simple solution to the stabilization problem when only position measurements are available. Thus elegant control laws composed of a potential energy compensation term and a linear dynamic (first order) compensator have been proposed very recently in [16,17]. It is noted that while these schemes eliminate the need for velocity measurements, they still require knowledge of the potential energy model for the robotic manipulator as well as any payload to provide accurate position control.

Research focusing on reducing the information required concerning the mechanical system potential energy has tended to concentrate on adaptive control approaches. The schemes presented in [18,19] possess a PD-plus-potential-model form in which only the structure of the potential energy model must be known; the (constant) parameters appearing in the model are assumed to be unknown, and this uncertainty is compensated adaptively. The information required concerning the potential energy is reduced further in the controller proposed in [20], where a scheme is given that adaptively compensates for uncertainty regarding both the structure and the parameter values of the system potential energy. It is mentioned that the adaptive controllers presented in [18-20] all possess a PD feedback component, and consequently all require velocity measurement. An additional observation concerning this work is that none of the stabilization schemes [16-20] have been verified through a comprehensive simulation and experimental study.

This paper introduces a new adaptive stabilization scheme for fully actuated mechanical systems that eliminates the need for any knowledge of the system dynamic model and that can be implemented using only measurements of the system configuration. It is shown that the control strategy guarantees semiglobal uniform boundedness of all signals and convergence of the system configuration error to zero. If bounded external disturbances are present, a slight modification to the control scheme ensures that uniform boundedness of all signals is retained and that arbitrarily accurate stabilization of the configuration error is achieved. The efficacy of the proposed controller is illustrated through both computer simulations and experiments involving three important manufacturing tasks: cutting force control in machining, control of contouring with a machine tool, and position control of an industrial robot.

The paper is organized as follows. In Section 2 some preliminary facts of relevance to the mechanical system stabilization problem are established. The adaptive stabilization scheme is presented and analyzed in Section 3. The performance of the controller is illustrated in Section 4 through a computer simulation study and in Section 5 through an experimental investigation. Finally, Section 6 summarizes the paper and draws some conclusions.

2. Preliminaries

Consider the class of mechanical systems that possess n configuration degrees-of-freedom (DOF) and $m \geq n$ actuators (of which n are independent). It can be shown that this class of systems can be modeled quantitatively as follows [21]:

$$M(\mathbf{x})\mathbf{T} = H(\mathbf{x})\ddot{\mathbf{x}} + V_{cc}(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}(\mathbf{x}) + \mathbf{d}(\mathbf{x},\dot{\mathbf{x}},t) \tag{1}$$

where $\mathbf{x} \in \Re^n$ is the vector of system generalized coordinates, $\mathbf{T} \in \Re^m$ is the vector of actuator inputs, $M \in \Re^{n \times m}$ is a bounded surjective matrix, $H \in \Re^{n \times n}$ is the system inertia matrix, $V_{cc} \in \Re^{n \times n}$ quantifies Coriolis and centripetal acceleration effects, and $\mathbf{G} \in \Re^n$ arises from the system potential energy. The term $\mathbf{d} \in \Re^n$ is an unknown vector, assumed to be bounded with bounded first derivative with respect to time, that can represent unmodelled state-dependent effects or time-dependent external disturbances; this term is included in the system dynamics to permit the robustness of the proposed stabilization scheme to be studied. It is well-known that the dynamics (1) possesses considerable structure [e.g., 21]. For example, it is easy to show that, for any set of generalized coordinates $\mathbf{x}$, the matrix H is symmetric and positive-definite, the matrix V_{cc} depends linearly on $\dot{\mathbf{x}}$, and the matrices H and V_{cc} are related according to $\dot{H} = V_{cc} + V_{cc}^T$ [21]. Additionally, we will assume in what follows that the inertia matrix H and potential energy gradient $\mathbf{G}$ are bounded with bounded first partial derivatives; this latter property holds for those mechanical systems of interest in manufacturing applications, such as robotic manipulators and machine tools.

Observe from (1) that the vector of actuator inputs $\mathbf{T}$ does not affect the system dynamics directly, and instead influences the system evolution through the column-space of M. By definition the columns of M span $\Re^n$, so that (1) is fully actuated and the situation is straightforward. However, it is very useful to explicitly identify the generalized force vector $\mathbf{F} \in \Re^n$ associated with the generalized coordinate vector $\mathbf{x}$:

$$\mathbf{F} = M(\mathbf{x})\mathbf{T} \tag{2}$$

so that (1) becomes:

$$\mathbf{F} = H(\mathbf{x})\ddot{\mathbf{x}} + V_{cc}(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}(\mathbf{x}) + \mathbf{d}(\mathbf{x},\dot{\mathbf{x}},t) \tag{3}$$

If $m > n$ then there exists an infinite number of actuator inputs corresponding to a given generalized force $\mathbf{F}$, which suggests that $\mathbf{T}$ can be chosen to enhance the performance of the system in some way. This observation and the partitioned representation of the system (2),(3) motivates the development of a control system that consists of two subsystems. The first subsystem is an adaptive scheme that generates the control input $\mathbf{F}$ required to stabilize the system (3), and the second subsystem is an algorithm for specifying the actuator input $\mathbf{T}$ that provides the desired $\mathbf{F}$ and simultaneously utilizes any available actuator redundancy to improve performance. Partitioning the control problem in this way is very natural and can be quite effective when applied to specific physical systems [22-24]. Given (2), an actuator input $\mathbf{T}$ corresponding to the desired control input $\mathbf{F}$ can always be determined; this problem is discussed extensively in [22-24] and the references therein. Therefore for the remainder of the paper it is assumed that the control input $\mathbf{F}$ can be commanded directly, and the focus of the subsequent discussion is on the problem of stabilizing the system (3) in the presence of incomplete information regarding this mechanical system. More specifically, the control objective is to specify the control input $\mathbf{F}$ to the system (3), using only measurements of $\mathbf{x}$ and without any *a priori* information concerning the system dynamic model, in such a way that the system (3) evolves from its initial state to the desired final state $\mathbf{x} = \mathbf{x}_d$, $\dot{\mathbf{x}} = \mathbf{0}$.

Our approach to the design and analysis of a suitable controller is based on Lyapunov stability theory. The following two lemmas summarize certain facts regarding a class of Lyapunov functions and will be of direct relevance in our development. The first lemma establishes a useful result concerning convergence of a dynamical system to a neighborhood of the origin.

Lemma 1: Consider the coupled dynamical system $\dot{\mathbf{x}}_1 = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, t)$, $\dot{\mathbf{x}}_2 = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, t)$. Let $V(\mathbf{x}_1, \mathbf{x}_2, t)$ be a Lyapunov function for the system with the properties

$$\lambda_1 \parallel \mathbf{x}_1 \parallel^2 + \lambda_2 \parallel \mathbf{x}_2 \parallel^2 \leq V \leq \lambda_3 \parallel \mathbf{x}_1 \parallel^2 + \lambda_4 \parallel \mathbf{x}_2 \parallel^2$$
$$\dot{V} \leq -\lambda_5 \parallel \mathbf{x}_1 \parallel^2 - \lambda_6 \parallel \mathbf{x}_2 \parallel^2 + \epsilon$$

where ϵ and the λ_i are positive scalar constants. Define $\delta = \max(\lambda_3/\lambda_5, \lambda_4/\lambda_6)$ and $r_i = (\delta\epsilon/\lambda_i)^{1/2}$ for $i = 1, 2$. Then for any initial state $\mathbf{x}_1(0), \mathbf{x}_2(0)$ the system will evolve so that $\mathbf{x}_1(t), \mathbf{x}_2(t)$ are uniformly bounded and converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where $B_{r_i} = \{\mathbf{x}_i \ : \ \parallel \mathbf{x}_i \parallel \leq r_i\}$ (see [25] for a discussion of exponential convergence to a closed ball).

Proof: The proof is based on an extension of the global exponential convergence theorem of Corless [25] and is given in [26]. □

The next result is a special case of LaSalle's invariant set theorem.

Lemma 2: Consider the coupled dynamical system $\dot{\mathbf{x}}_1 = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2)$, $\dot{\mathbf{x}}_2 = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2)$ with equilibrium at the origin. Let $V(\mathbf{x}_1, \mathbf{x}_2)$ be a Lyapunov function for the system with the properties

$$\lambda_1 \parallel \mathbf{x}_1 \parallel^2 + \lambda_2 \parallel \mathbf{x}_2 \parallel^2 \leq V \leq \lambda_3 \parallel \mathbf{x}_1 \parallel^2 + \lambda_4 \parallel \mathbf{x}_2 \parallel^2$$
$$\dot{V} \leq -\lambda_5 \parallel \mathbf{x}_1 \parallel^2$$

where the λ_i are positive scalar constants. Then the origin is a stable equilibrium, and $\mathbf{x}_1 \to \mathbf{0}$ as $t \to \infty$.

Proof: The proof can be found in [e.g., 27]. □

3. Adaptive Stabilization Scheme

We now turn to the development of the proposed approach to stabilizing uncertain mechanical systems using only configuration measurements. Before presenting our main results, we briefly consider the nonadaptive case to fix some ideas and introduce a useful Lyapunov function.

3.1 Nonadaptive Case

Consider the problem of stabilizing the mechanical system (3) using only measurements of system configuration but with complete knowledge of the system dynamic model. In [16], Berghuis and Nijmeijer proposed the following elegant solution to this problem for robotic manipulators based on the seminal work of Takegaki and Arimoto [7]:

$$\mathbf{F} = \mathbf{G}(\mathbf{x}) + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \qquad (4)$$
$$\dot{\mathbf{w}} = -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}}$$

where $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$ is the position regulation error (recall that $\mathbf{x}_d$ is constant), k_1, k_2, γ are positive scalar constants, and where we have reformulated the scheme given in [16] somewhat to make our subsequent analysis more convenient. Note that (4) is implementable without velocity information because, although $\dot{\mathbf{w}}$ depends on $\dot{\mathbf{e}}$, the control law requires only $\mathbf{w}$ and this term depends only on $\mathbf{e}$. It is shown in [16] that (4) globally stabilizes the dynamics (3) at the equilibrium $\mathbf{e} = \dot{\mathbf{e}} = \mathbf{w} = \mathbf{0}$; however, the stability analysis in [16] employs a Lyapunov function with a negative-semidefinite time derivative, so that it is not clear how to extend this analysis to the adaptive case. In what follows, we present an alternative stability analysis which demonstrates that the equilibrium point is exponentially stable, and which forms the basis for our derivation of an adaptive solution to the problem.

Lemma 3: The controller (4) exponentially stabilizes the *nominal* mechanical system dynamics (3) (with $\mathbf{d} = \mathbf{0}$) at the equilibrium $\mathbf{e} = \dot{\mathbf{e}} = \mathbf{w} = \mathbf{0}$ provided γ is chosen sufficiently large.

Proof: Applying the controller (4) to the nominal mechanical system dynamics (3) yields the closed-loop system

$$\begin{aligned} H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} &= 0 \\ \dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \end{aligned} \tag{5}$$

Consider the Lyapunov function candidate

$$V_1 = \frac{1}{2}\dot{\mathbf{e}}^T H\dot{\mathbf{e}} + \frac{1}{2}k_2\gamma^2\mathbf{e}^T\mathbf{e} + \frac{1}{2}k_1\mathbf{w}^T\mathbf{w} + \frac{k_2}{k_1\gamma}\mathbf{e}^T H\dot{\mathbf{e}} - \frac{1}{\gamma}\mathbf{w}^T H\dot{\mathbf{e}} \tag{6}$$

and note that V_1 in (6) is positive-definite and radially-unbounded provided γ is chosen large enough. Differentiating (6) along (5) and simplifying yields

$$\begin{aligned} \dot{V}_1 &= -(\gamma - \frac{k_2}{k_1\gamma})\dot{\mathbf{e}}^T H\dot{\mathbf{e}} - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 + 2\mathbf{w}^T H\dot{\mathbf{e}} \\ &+ \frac{1}{\gamma}\dot{\mathbf{e}}^T[\frac{k_2}{k_1}V_{cc}\mathbf{e} - V_{cc}\mathbf{w}] \\ &\leq -(\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\ &+ 2\lambda_{max}(H) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 \end{aligned} \tag{7}$$

where $\lambda_{min}(\cdot), \lambda_{max}(\cdot)$ denote the minimum and maximum eigenvalue of the matrix argument, respectively, and k_{cc} is a scalar upper bound on the linear dependency of V_{cc} on $\dot{\mathbf{x}}$ (i.e., $\parallel V_{cc} \parallel_F \leq k_{cc} \parallel \dot{\mathbf{x}} \parallel \forall \mathbf{x}$ with $\parallel \cdot \parallel_F$ the Frobenius matrix norm). Define $\mathbf{z} = [\parallel \mathbf{e} \parallel \;\; \parallel \dot{\mathbf{e}} \parallel \;\; \parallel \mathbf{w} \parallel]^T$ and

$$Q_1 = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} & 0 & 0 \\ 0 & (\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) & -\lambda_{max}(H) \\ 0 & -\lambda_{max}(H) & k_1\gamma \end{bmatrix}$$

and note that Q_1 is positive-definite if γ is chosen sufficiently large. The following bound on $\dot{V}_1$ in (7) can then be established:

$$\begin{aligned} \dot{V}_1 &\leq -\mathbf{z}^T Q_1\mathbf{z} + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 \\ &\leq -(\lambda_{min}(Q_1) - \frac{\alpha_1}{\gamma}{V_1}^{1/2}) \parallel \mathbf{z} \parallel^2 \end{aligned} \tag{8}$$

for some scalar constant α_1 which does not increase as γ is increased. Let $c_1(t) = \lambda_{min}(Q_1) - \alpha_1 {V_1}^{1/2}(t)/\gamma$ and choose γ large enough so that $c_1(0) > 0$ (this is always possible). Then (8) implies that $c_1(0) \leq c_1(t) \quad \forall t$, so that $\dot{V}_1 \leq -c_1(0) \parallel \mathbf{z} \parallel^2$. Lemma 2 (with $\epsilon = 0$) then permits the conclusion that $\mathbf{e}, \dot{\mathbf{e}}, \mathbf{w}$ converge exponentially to zero. □

Observe that in the preceding stability analysis a negative-definite Lyapunov function derivative has been obtained. It will be shown in what follows that, as a consequence, this analysis can be readily extended to the adaptive case. Note also that the stability properties established in Lemma 3 are *semiglobal* in the sense that the region of attraction can be increased arbitrarily by increasing the controller gain γ. It is stressed that this does not imply that γ must be chosen to be overly large; indeed, we have found that excellent performance is obtainable with this gain set to quite modest values.

3.2 Adaptive Stabilization

As indicated above, it is desirable to eliminate the need for information concerning the potential energy model in mechanical system stabilization schemes. Consider the following adaptive approach to realizing this objective:

$$\begin{aligned} \mathbf{F} &= \mathbf{f}(t) + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \\ \dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \\ \dot{\mathbf{f}} &= \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}) \end{aligned} \tag{9}$$

where $\mathbf{f}(t) \in \Re^n$ is the adaptive element and β is a positive scalar constant; it is clear that the adaptive term $\mathbf{f}(t)$ is intended to compensate for potential energy effects. Observe that (9) is implementable without velocity information because, although $\dot{\mathbf{w}}$ and $\dot{\mathbf{f}}$ depends on $\dot{\mathbf{e}}$, the control law terms $\mathbf{w}$ and $\mathbf{f}$ depend only on $\mathbf{e}$ (as can be verified through direct integration of the $\dot{\mathbf{w}}$ and $\dot{\mathbf{f}}$ equations). Applying the control law (9) to the *nominal* mechanical system dynamics (3) (with $\mathbf{d} = \mathbf{0}$) yields the closed-loop error dynamics

$$\begin{aligned} H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi} + \mathbf{G}(\mathbf{x}_d) - \mathbf{G}(\mathbf{x}) &= \mathbf{0} \\ \dot{\mathbf{w}} = -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} & \\ \dot{\mathbf{f}} = \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}) & \end{aligned} \tag{10}$$

where $\mathbf{\Phi} = \mathbf{f} - \mathbf{G}(\mathbf{x}_d)$.

The stability properties of the proposed adaptive stabilization strategy (9) are summarized in the following theorem.

Theorem 1: The adaptive scheme (9) ensures that origin $\mathbf{e} = \dot{\mathbf{e}} = \mathbf{w} = \mathbf{\Phi} = \mathbf{0}$ is a semiglobally stable equilibrium point for (10), and that $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$ converge to zero asymptotically, provided γ is chosen sufficiently large.

Proof: Let $U(\mathbf{x})$ denote the potential energy of the mechanical system (3). Consider the Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2\beta}\mathbf{\Phi}^T\mathbf{\Phi} + U(\mathbf{x}) - U(\mathbf{x}_d) + \mathbf{G}^T(\mathbf{x}_d)\mathbf{e} \tag{11}$$

and note that V_2 is a positive-definite and radially-unbounded function of $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$ and $\mathbf{\Phi}$ if γ is chosen sufficiently large (for example, if γ is chosen large enough then the term $k_2\gamma^2\mathbf{e}^T\mathbf{e}/2 + U(\mathbf{x}) - U(\mathbf{x}_d) + \mathbf{G}^T(\mathbf{x}_d)\mathbf{e}$ can be shown to be positive-definite in $\mathbf{e}$ using an analysis analogous to the one given in [18]).

Computing the derivative of (11) along (10) and simplifying as in the proof of Lemma 3 yields

$$\begin{aligned}
\dot{V}_2 \leq & -(\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\
& + 2\lambda_{max}(H) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2 \\
& + \frac{1}{\beta}\mathbf{\Phi}^T(\dot{\mathbf{f}} - \beta[\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}]) + \parallel \mathbf{G}(\mathbf{x}_d) - \mathbf{G}(\mathbf{x}) \parallel\parallel \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w} \parallel \\
\leq & -(\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - (\frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma}) \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\
& + 2\lambda_{max}(H) \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel + \frac{M}{\gamma} \parallel \mathbf{w} \parallel\parallel \mathbf{e} \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel\parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel\parallel \dot{\mathbf{e}} \parallel^2
\end{aligned} \tag{12}$$

where M is a positive scalar constant satisfying $M \parallel \mathbf{x}_d - \mathbf{x} \parallel \geq \parallel \mathbf{G}(\mathbf{x}_d) - \mathbf{G}(\mathbf{x}) \parallel \forall \mathbf{x}_d, \mathbf{x}$ (the boundedness of the partial derivatives of $\mathbf{G}$ ensures that such an M exists). Define $\mathbf{z}$ as in Lemma 3 and Q_2 as

$$Q_2 = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} - \frac{k_2 M}{k_1\gamma} & 0 & -\frac{M}{2\gamma} \\ 0 & (\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) & -\lambda_{max}(H) \\ -\frac{M}{2\gamma} & -\lambda_{max}(H) & k_1\gamma \end{bmatrix}$$

and note that Q_2 is positive-definite if γ is chosen large enough. Then the following bound on $\dot{V}_2$ in (12) can be obtained through routine manipulation:

$$\dot{V}_2 \leq -(\lambda_{min}(Q_2) - \frac{\alpha_2}{\gamma}{V_2}^{1/2}) \parallel \mathbf{z} \parallel^2 \tag{13}$$

where α_2 is a scalar constant which does not increase as γ is increased. Let $c_2(t) = \lambda_{min}(Q_2) - \alpha_2 {V_2}^{1/2}(t)/\gamma$ and choose γ large enough so that $c_2(0) > 0$. Then (13) implies that $c_2(0) \leq c_2(t) \;\; \forall t$, so that $\dot{V}_2 \leq -c_2(0) \parallel \mathbf{z} \parallel^2$. Lemma 2 then permits the conclusion that $\mathbf{e} = \dot{\mathbf{e}} = \mathbf{w} = \mathbf{\Phi} = \mathbf{0}$ is (semiglobally) stable and that $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$ converge to zero asymptotically. □

Several observations can be made concerning the adaptive stabilization strategy (9). First, the preceding analysis demonstrates that this scheme ensures that the nominal mechanical system dynamics (3) evolves from any initial state to any desired final configuration with bounded signals and with convergence of the state error to zero. The controller requires no *a priori* information concerning the system model and is implementable using only configuration measurements. Thus the proposed scheme provides a computationally efficient, modular, and readily implementable solution to the mechanical system stabilization problem. An additional observation is that the scalar controller gains k_1, k_2, γ, and β can be replaced with the appropriate diagonal matrices for increased flexibility and performance; scalar gains are used in the above analysis for simplicity of development. Finally, it is interesting to note that the proposed controller (9) can be readily modified to provide increased robustness to unmodeled effects and external disturbances; this modification is considered next.

3.3 Robust Adaptive Stabilization

Consider the following modification of the adaptive stabilization scheme (9):

$$\begin{aligned}
\mathbf{F} &= \mathbf{f}(t) + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} \\
\dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \\
\dot{\mathbf{f}} &= -\sigma\mathbf{f} + \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w})
\end{aligned} \tag{14}$$

where σ is a positive scalar constant. Applying the controller (14) to the ***disturbed*** mechanical system dynamics (3) (with $\mathbf{d}$ nonzero) yields the closed-loop error dynamics

$$\begin{aligned}
H\ddot{\mathbf{e}} + V_{cc}\dot{\mathbf{e}} + k_1\gamma^2\mathbf{w} + k_2\gamma^2\mathbf{e} + \mathbf{\Phi}_d &= \mathbf{0} \\
\dot{\mathbf{w}} &= -2\gamma\mathbf{w} + \gamma^2\dot{\mathbf{e}} \\
\dot{\mathbf{f}} &= -\sigma\mathbf{f} + \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w})
\end{aligned} \tag{15}$$

where $\mathbf{\Phi}_d = \mathbf{f} - \mathbf{G} - \mathbf{d}$.

The stability properties of the proposed adaptive stabilization strategy (14) are summarized in the following theorem.

Theorem 2: The adaptive scheme (14) ensures that $\mathbf{e}, \dot{\mathbf{e}}, \mathbf{w}, \mathbf{\Phi}_d$ are globally uniformly bounded provided that γ is chosen sufficiently large. Moreover, the stabilization error $\mathbf{e}, \dot{\mathbf{e}}$

is guaranteed to converge exponentially to a compact set which can be made arbitrarily small.

Proof: Consider the Lyapunov function candidate

$$V_3 = V_1 + \frac{1}{2\beta}\mathbf{\Phi}_d^T\mathbf{\Phi}_d \tag{16}$$

and note that V_3 is a positive-definite and radially-unbounded function of $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$ and $\mathbf{\Phi}_d$ if γ is chosen sufficiently large. Differentiating (16) along (15) and simplifying as in the proof of Lemma 3 yields

$$\begin{aligned}\dot{V}_3 \leq & -(\gamma - \frac{k_2}{k_1\gamma})\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{k_2^2\gamma}{k_1} \parallel \mathbf{e} \parallel^2 - k_1\gamma \parallel \mathbf{w} \parallel^2 \\ & + 2\lambda_{max}(H) \parallel \mathbf{w} \parallel \parallel \dot{\mathbf{e}} \parallel + \frac{k_2 k_{cc}}{k_1\gamma} \parallel \mathbf{e} \parallel \parallel \dot{\mathbf{e}} \parallel^2 + \frac{k_{cc}}{\gamma} \parallel \mathbf{w} \parallel \parallel \dot{\mathbf{e}} \parallel^2 \\ & + \frac{1}{\beta}\mathbf{\Phi}_d^T(\dot{\mathbf{f}} - \dot{\mathbf{G}} - \dot{\mathbf{d}} - \beta(\dot{\mathbf{e}} + \frac{k_2}{k_1\gamma}\mathbf{e} - \frac{1}{\gamma}\mathbf{w}))\end{aligned} \tag{17}$$

Defining $\mathbf{z}$ as in Lemma 3 and Q_3 as

$$Q_3 = \begin{bmatrix} \frac{k_2^2\gamma}{k_1} & 0 & 0 \\ 0 & (\frac{\gamma}{2} - \frac{k_2}{k_1\gamma})\lambda_{min}(H) & -\lambda_{max}(H) \\ 0 & -\lambda_{max}(H) & k_1\gamma \end{bmatrix}$$

(and noticing that Q_3 is positive-definite if γ is chosen large enough) permits the following bound on $\dot{V}_3$ in (17) to be obtained:

$$\begin{aligned}\dot{V}_3 & \leq -(\lambda_{min}(Q_3) - \frac{\alpha_3}{\gamma}{V_3}^{1/2}) \parallel \mathbf{z} \parallel^2 - \frac{\sigma}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2 \\ & \quad - \frac{\gamma}{2}\lambda_{min}(H) \parallel \dot{\mathbf{e}} \parallel^2 - \frac{\sigma}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2 + \frac{\alpha_4}{\beta} \parallel \mathbf{\Phi}_d \parallel \parallel \dot{\mathbf{e}} \parallel + \frac{\alpha_5}{\beta} \parallel \mathbf{\Phi}_d \parallel \\ & \leq -(\lambda_{min}(Q_3) - \frac{\alpha_3}{\gamma}{V_3}^{1/2}) \parallel \mathbf{z} \parallel^2 - \frac{\sigma}{2\beta} \parallel \mathbf{\Phi}_d \parallel^2 + \frac{\alpha_6}{\beta}\end{aligned} \tag{18}$$

where the α_i are scalar constants which do not increase as γ, β increase.

The remainder of the proof will utilize the result given in Lemma 1, and therefore it is convenient to bound V_3 in (16) in the following manner:

$$\lambda_1 \parallel \mathbf{z} \parallel^2 + \frac{\lambda_2}{\beta} \parallel \mathbf{\Phi}_d \parallel^2 \leq V_3 \leq \lambda_3 \parallel \mathbf{z} \parallel^2 + \frac{\lambda_2}{\beta} \parallel \mathbf{\Phi}_d \parallel^2$$

where the λ_i are positive scalar constants independent of β. Now choose two scalar constants V_M, V_m so that $V_M > V_m \geq V_3(0)$, and define $c_M = \lambda_{min}(Q_3) - \alpha_3 {V_M}^{1/2}/\gamma$; then choose γ large enough so that $c_M > 0$ (this is always possible). Let $\delta = \max(\lambda_3/c_M, 2\lambda_2/\sigma)$ and choose β_0 so that $\alpha_6\delta/\beta_0 < V_m$ (this is always possible). Then selecting $\beta \geq \beta_0$ ensures that if $V_m \leq V_3 \leq V_M$ then $\dot{V}_3 < 0$. This condition together with $V_M > V_m \geq V_3(0)$ implies that $V_3(t) \leq V_M \; \forall t$, so that $c_3(t) = \lambda_{min}(Q_3) - \alpha_3 {V_3}^{1/2}(t)/\gamma > c_M > 0 \;\; \forall t$ and

$$\dot{V}_3 \leq -c_M \parallel \mathbf{z} \parallel^2 - \frac{\sigma}{2(\beta_0 + \Delta\beta)} \parallel \mathbf{\Phi}_d \parallel^2 + \frac{\alpha_6}{(\beta_0 + \Delta\beta)}$$

where $\Delta\beta = \beta - \beta_0$ and it is assumed that β is chosen so that $\Delta\beta > 0$. Lemma 1 now applies and permits the conclusion that $\parallel \mathbf{z} \parallel, \parallel \mathbf{\Phi}_d \parallel$ are uniformly bounded, which implies that $\mathbf{e}$, $\dot{\mathbf{e}}$, $\mathbf{w}$, $\mathbf{f}$ are uniformly bounded. Moreover, $\parallel \mathbf{z} \parallel, \parallel \mathbf{\Phi}_d \parallel$ converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where

$$r_1 = \left(\frac{\delta\alpha_6}{\lambda_1(\beta_0 + \Delta\beta)} \right)^{1/2}$$

$$r_2 = \left(\frac{\delta\alpha_6}{\lambda_2} \right)^{1/2}$$

Observe that the radius of the ball to which $\parallel \mathbf{z} \parallel^2 = \parallel \mathbf{e} \parallel^2 + \parallel \dot{\mathbf{e}} \parallel^2 + \parallel \mathbf{w} \parallel^2$ is guaranteed to converge can be decreased as desired simply by increasing $\Delta\beta$. □

Several observations can be made concerning the adaptive stabilization strategy (14). First, the preceding analysis demonstrates that this scheme ensures that the *disturbed* mechanical system dynamics (3) evolves from any initial state to any desired final configuration with uniform boundedness of all signals and with exponential convergence of the state error to an arbitrarily small neighborhood of the origin. Note that the ultimate size of the regulation error can be reduced as desired simply by increasing a single controller parameter (the adaptation gain β), and that exponential convergence ensures that the transient behavior of the errors is well-behaved. The controller does not require knowledge of the structure or parameter values for the sytem model, and is implementable using only configuration measurements. Thus the proposed scheme provides a computationally efficient, modular, readily implementable, and robust solution to the mechanical system stabilization problem. An additional observation is that the scalar controller gains k_1, k_2, γ, σ, and β can be replaced with the appropriate diagonal matrices for increased flexibility and performance; scalar gains are used in the above analysis for simplicity of development.

4. Case Studies: Simulation Results

This Section describes the results of a computer simulation study involving the application of the proposed adaptive stabilization strategy (14) to two important machine tool control problems: cutting force control in milling and contouring control in machining. In each of these applications the plant dynamics can be modeled in the general form (2),(3), and in each case there is considerable motivation for seeking a stabilization scheme which does not require rate measurements or specific model information. Throughout the study the unit of time is second, the unit of force is Newton, and the unit of length is millimeter, unless stated otherwise.

4.1 Cutting Force Control in Milling

It has been recognized that adaptively controlling the cutting forces and torques encountered during machining operations could permit material removal rates to be increased while maintaining safe machining conditions [e.g, 28,29]. Previous approaches to designing adaptive controllers for this application have usually been based on linearized models of the milling process and have often assumed that force rate information is available (which is not realistic). As a consequence, there has been concern regarding the stability of the schemes and the application of these control systems has been limited. In this example we consider the problem of controlling the cutting force in milling by manipulating the feedrate override voltage using only measurements of cutting force. The work reported in [28,29] shows that the plant associated with this control problem is a mechanical system of the general form (2),(3) with $m = n = 1$; the precise form of the model is strongly dependent on the machine tool and workpiece and it is usually not realistic to assume that this model is known with any precision.

This simulation examines the capability of the proposed stabilization scheme (14) to accurately and robustly control cutting force in machining. The simulation utilizes the cutting force model developed in [28,29], and all model and task parameters are as given in [29]. It is stressed that this model information is used only for computer simulation of the system and is not made available to the controller. The cutting tool is initially not in contact with the workpiece, and the tool is requested to exert a cutting force of 500N on the workpiece; thus the cutting tool is driven to the workpiece under the control of the proposed stabilization scheme. Then, midway through the cut (at $t = 5.0$s), the depth of cut is suddenly reduced by a factor of four, causing an abrupt change in the system model (which is sensitive to changes in depth of cut). The adaptive gain $\mathbf{f}$ is initialized to zero, and the remaining controller parameters are set as follows: $k_1 = 10$, $k_2 = 10$, $\gamma = 5$, $\sigma = 0.01$, $\beta = 1000$. The adaptive controller is applied to the milling machine with a sampling period of two milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of two milliseconds. The results of this simulation are given in Figure 1, and indicate that accurate cutting force regulation is achieved with the proposed scheme (14). Additionally, this simulation shows that the controller is robust to the abrupt and unexpected change in the system dynamics which occurs when the depth of cut is suddenly reduced midway through the task.

4.2 Contouring with Machine Tools

The contouring problem for multiaxis machine tools involves controlling the motion of the cutting tool relative to the workpiece so that the tool accurately tracks a given trajectory [e.g., 30,31]. The controllers which are presently employed for this application are typically designed using classical techniques under the assumption that the system is linear, and one drawback with this approach is that tracking accuracy degrades if the desired trajectory requires fast motions or involves tight contours. Another potential problem with these schemes is their reliance on velocity information, since measurements of velocity are often contaminated with noise [12]. Here we consider implementing the proposed stabilization scheme (14) for contouring control with a two-axis milling machine. The investigations [30,31] show that the plant associated with this control problem is a mechanical system of the general form (2),(3) with $m = n = 2$; the details of the model depend on the machine tool and workpiece and are not likely to be known with any precision.

This simulation examines the capability of the proposed stabilization scheme (14) to accurately track a given contouring trajectory with a two-axis milling machine. Note that, although the controller (14) is not designed for tracking, it can be expected that the adaptive nature of the scheme should permit good tracking accuracy to be obtained. This simulation utilizes the machine tool model presented in [31], and all model and task parameters are as given in [31]. Again, it is stressed that this model information is used only to simulate the dynamical system and is not made available to the controller. As indicated above, a potential advantage of this approach to mechanical system control is that velocity information, which is typically noisy, is not required by the controller. To examine this potential, the proposed scheme (14) is implemented both with and without velocity measurements (in the case that velocity is available, this information is simply used in place of $\mathbf{w}$ in (14)). The velocity information is modeled as a superposition of the true velocity and a measurement error component, where the error is assumed to be a zero-mean Gaussian signal with a standard deviation equal to one percent of the nominal signal; thus the measurements are modeled as being relatively free of error.

In this simulation, the milling machine is commanded to track a circular contour of radius 100mm in 10.0s, and to initiate and terminate the contouring with zero velocity. The adaptive gain $\mathbf{f}$ is initialized to zero, and the remaining controller parameters are set as follows: $k_1 = 1000$, $k_2 = 1000$, $\gamma = 5$, $\sigma = 0.001$, $\beta = 10000$. The adaptive controller is applied to the milling machine with a sampling period of two milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of two milliseconds. The results of this simulation are given in Figure 2, and indicate that accurate trajectory tracking is achieved with the proposed scheme (14). Additionally, this simulation shows that the controller is capable of providing performance superior to that obtainable with a state-feedback adaptive controller if there is even a modest level of noise in the velocity measurements.

5. Case Studies: Experimental Results

The proposed adaptive stabilization scheme (14) is now applied to an industrial robot in a series of hardware experiments. The system chosen for this study is the IMI Zebra Zero manipulator (see Figure 3). This robot is an electrically-driven six DOF revolute joint arm, and is composed of a trunk, shoulder, upper arm, forearm, wrist, and end-effector. The selection of this type of system for study is motivated by the fact that robots form an important and well-known class of mechanical systems, and that there is a recognized need for high-performance control of these systems in the presence of uncertainty and disturbances. In what follows some familiarity with robots is assumed, and the interested reader is referred to [32] for additional details regarding these systems. Throughout the study the unit of time is second and the unit of angle is degree.

5.1 Position Regulation of Industrial Robots

The adaptive stabilization strategy (14) is now applied to an industrial robot in an experimental investigation. The experimental facility utilized for this study is the New Mexico State University Robotics Laboratory and consists of a IMI Zebra Zero robot arm and the associated control computer (see Figure 3). The Zebra robot possesses a conventional design with six revolute joints configured in a "waist-shoulder-elbow-wrist" arrangement, and the system dynamic model is of the general form (2),(3) with $m = n = 6$; the precise form of the model depends on the particular choice of generalized coordinates. All control software is written in 'C' and is hosted on an IBM-compatible 486 personal computer. A complete description of this flexible and modular experimental testbed is beyond the scope of this paper; the interested reader is referred to [33] for such a description. It is noted, however, that the open architecture environment is designed to allow convenient implementation and experimental evaluation of advanced control schemes using the Zebra robot, and therefore provides an ideal testbed for the present investigation.

The first experiment in this case study demonstrates the capability of the proposed adaptive scheme (14) to provide accurate position regulation. The Zebra manipulator is initially at rest with joint-space position $\theta(0) = [-45°, 0°, -100°, 0°, 45°, 0°]^T$, and is commanded to move to the desired final configuration $\theta_d = [15°, 60°, -40°, 60°, 105°, 60°]^T$. Thus each joint is requested to move through 60° and to initiate and terminate the motion with zero velocity. The adaptive gain $\mathbf{f}$ is initialized to zero, and the remaining controller parameters are set as follows: $k_1 = 10$, $k_2 = 20$, $\gamma = 5$, $\sigma = 0.001$, $\beta = 100$. The adaptive controller is applied to the robot with a sampling period of seven milliseconds, and all integrations required by the controller are implemented using a simple trapezoidal integration rule with a time-step of seven milliseconds. The results of this experiment are given in Figures 4a and 4b, and indicate that accurate position regulation is achieved with the proposed scheme (14).

5.2 Trajectory Tracking of Industrial Robots

In the next experiment, the adaptive stabilization scheme (14) is applied to a robot trajectory tracking problem. While utilizing stabilization control systems for tracking is not new in robotics, this experiment indicates that the capacity for adaptation can permit even rapid trajectories to be tracked with good accuracy. The Zebra manipulator is again initially at rest with the same joint-space position as in the first experiment. The waist joint is commanded to smoothly move from its initial position of $\theta_1 = -45°$ to an intermediate position of $\theta_1 = 45°$, and then to return to its initial position; the desired temporal trajectory for this motion is specified to be $\theta_{1d} = -45° + 45°(1 - \cos(\pi/2)t)$ for $t \in [0, 4]$. At the same time, the shoulder joint is commanded to move from $\theta_2 = 0°$ to $\theta_2 = 57.3°$ (1 radian) and then back to $\theta_2 = 0°$ according to the trajectory $\theta_{2d} = 28.7°(1 - \cos(\pi/2)t)$ for $t \in [0, 4]$. All of the other manipulator joints are commanded to remain in their initial positions. Thus the manipulator is required to track a large and rapid joint-space trajectory with no *a priori* information concerning the system model. The control strategy (14) is utilized to achieve the specified trajectory tracking, and the sampling period and all controller parameters are set to the values used in the first experiment. The results of this experiment are given in Figures 5a and 5b, and show that accurate trajectory tracking is achieved.

6. Conclusions

This paper introduces a new adaptive stabilization scheme for mechanical systems that eliminates the need for any knowledge of the system dynamic model and that can be implemented using only measurements of the system configuration. The control strategy is computationally efficient, modular, and readily implementable with a wide variety of systems. It is demonstrated through computer simulations and experiments that the proposed controller can be applied to manufacturing tasks of practical importance, including cutting force control in machining, control of contouring with a machine tool, and position control of an industrial robot. In the cutting force control example, the scheme is applied to obtain accurate and robust force control using only force measurements and without any *a priori* information about the machine tool or workpiece. The applications involving the machine tool and the robot illustrate that the scheme can be used for both position regulation and trajectory tracking, and confirm that accurate motion control can be achieved without model information or rate measurements. Future research will focus on the application of the proposed approach to the problem of controlling constrained mechanical systems.

7. Acknowledgments

The research described in this paper was supported in part through contracts with the Army Research Office and Sandia National Laboratories.

8. References

1. Marino, R., "Stabilization and Feedback Equivalence to Linear Coupled Oscillators", *International Journal of Control*, Vol. 39, 1984, pp. 487-496

2. Schaft, A. van der, "Stabilization of Hamiltonian Systems", *Nonlinear Analysis: Theory, Methods, and Applications*, Vol. 10, No. 10, 1986, pp. 1021-1035

3. Koditschek, D., "The Control of Natural Motion in Mechanical Systems", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, No. 4, 1991, pp. 547-551

4. Aeyels, D., "On Stabilization by Means of the Energy-Casimir Method", *Systems and Control Letters*, Vol. 18, 1992, pp. 325-328

5. Bloch, A., M. Reyhanoglu, and N.H. McClamroch, "Control and Stabilization of Nonholonomic Dynamic Systems", *IEEE Transactions on Automatic Control*, Vol. 37, No. 11, 1992, pp. 1746-1757

6. Abesser, H. and M. Katzschmann, "Structure-Preserving Stabilization of Hamiltonian Control Systems", *Systems and Control Letters*, Vol. 22, 1994, pp. 281-285

7. Takegaki, M. and S. Arimoto, "A New Feedback Method for Dynamic Control of Manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, 1981, pp. 119-125

8. Paden, B. and R. Panja, "Globally Asymptotically Stable 'PD+' Controller for Robot Manipulators", *International Journal of Control*, Vol. 47, No. 6, 1988, pp. 1697-1712

9. McClamroch, N.H. and D. Wang, "Feedback Stabilization and Tracking of Constrained Robots", *IEEE Transactions on Automatic Control*, Vol. 33, No. 5, 1988, pp. 419-426

10. Crouch, P., "Spacecraft Attitude Control and Stabilization: Application of Geometrical Control Theory to Rigid Body Models", *IEEE Transactions on Automatic Control*, Vol. 29, No. 4, 1984, pp. 321-331

11. Byrnes, C. and A. Isidori, "On the Attitude Stabilization of Rigid Spacecraft", *Automatica*, Vol. 27, No. 1, 1991, pp. 87-95

12. Klafterm, R., T. Chmielewski, and M. Negin, *Robotic Engineering: An Integrated Approach*, Prentice Hall, Englewood Cliffs, NJ, 1989

13. Nicosia, S. and P. Tomei, "Robot Control by Using Only Joint Position Measurements", *IEEE Transactions on Automatic Control*, Vol. 35, No. 9, 1990, pp. 1058-1061

14. Canudas de Wit, C. and N. Fixot, "Robot Control via Robust Estimated State Feedback", *IEEE Transactions on Automatic Control*, Vol. 36, No. 12, 1991, pp. 1497-1501

15. Berghuis, H. and H. Nijmeijer, "A Passivity Approach to Controller-Observer Design for Robots" *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 6, 1993, pp. 740-754

16. Berghuis, H. and H. Nijmeijer, "Global Regulation of Robots Using Only Position Measurements" *Systems and Control Letters*, Vol. 21, 1993, pp. 289-293

17. Feng, W. and I. Postlethwaite, "A Simple Robust Control Scheme for Robot Manipulators With Only Joint Position Measurements", *International Journal of Robotics Research*, Vol. 12, No. 5, 1993, pp. 490-496

18. Tomei, P., "Adaptive PD Controller for Robot Manipulators", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 565-570

19. Kelly, R., "Comments on 'Adaptive PD Controller for Robot Manipulators' ", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 1, 1993, pp. 117-119

20. Colbaugh, R., "Adaptive Point-to-Point Motion Control of Manipulators", *International Journal of Robotics and Automation*, Vol. 9, No. 2, 1994, pp. 51-61

21. Saletan, E. and A. Cromer, *Theoretical Mechanics*, Wiley, New York, 1971

22. Colbaugh, R. and K. Glass, "On Controlling Robots with Redundancy", *Robotics and Computer-Integrated Manufacturing: An International Journal*, Vol. 9, No. 2, 1992, pp. 121-135

23. Colbaugh, R. and K. Glass, "Hierarchical Control of Human Joint Motion Simulators", *Computers and Electrical Engineering: An International Journal*, Vol. 19, No. 3, 1993, pp. 213-230

24. Colbaugh, R., K. Glass, and P. Pittman, "Adaptive Control for a Class of Hamiltonian Systems", *Computers and Electrical Engineering: An International Journal*, Vol. 20, No. 1, 1994, pp. 21-38

25. Corless, M., "Guaranteed Rates of Exponential Convergence for Uncertain Systems", *Journal of Optimization Theory and Applications*, Vol. 64, No. 3, 1990, pp. 481-494

26. Colbaugh, R., H. Seraji, and K. Glass, "Adaptive Compliant Motion Control for Dexterous Manipulators", *International Journal of Robotics Research*, Vol. 14, No. 3, 1995, pp. 270-280

27. Vidyasagar, M., *Nonlinear Systems Analysis*, Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1993

28. Lauderbaugh, L. and A. Ulsoy, "Model Reference Adaptive Control in Milling", *ASME Journal of Engineering for Industry*, Vol. 111, No. 1, 1989, pp. 13-21

29. Sullivan, G., "Adaptive Control with an Expert System Based Supervisory Level",

CIRSSE Report 115, Rensselaer Polytechnic Institute, 1991

30. Chuang, H. and C. Liu, “Cross-Coupled Adaptive Feedrate Control for Multiaxis Machine Tools”, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, No. 3, 1991, pp. 451-457

31. Lo, C.-C. and Y. Koren, “Evaluation of Servo-Controllers for Machine Tools”, *Proc. American Control Conference*, Chicago, IL, June 1992

32. Spong, M. and M. Vidyasagar, *Robot Dynamics and Control*, Wiley, New York, 1989

33. Glass, K. and R. Colbaugh, “A Robust Open-Architecture Experimental Testbed for Manipulator Controller Development”, Robotics Laboratory Report 94-01, New Mexico State University, February 1994

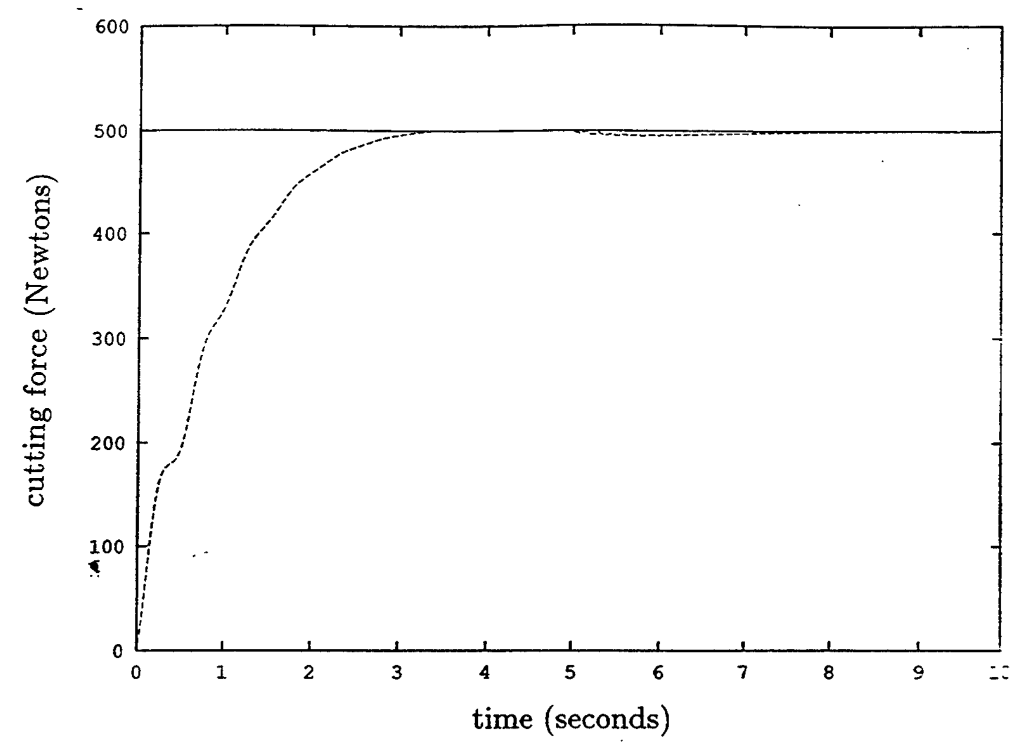


Figure 1: Response of cutting force in milling simulation study

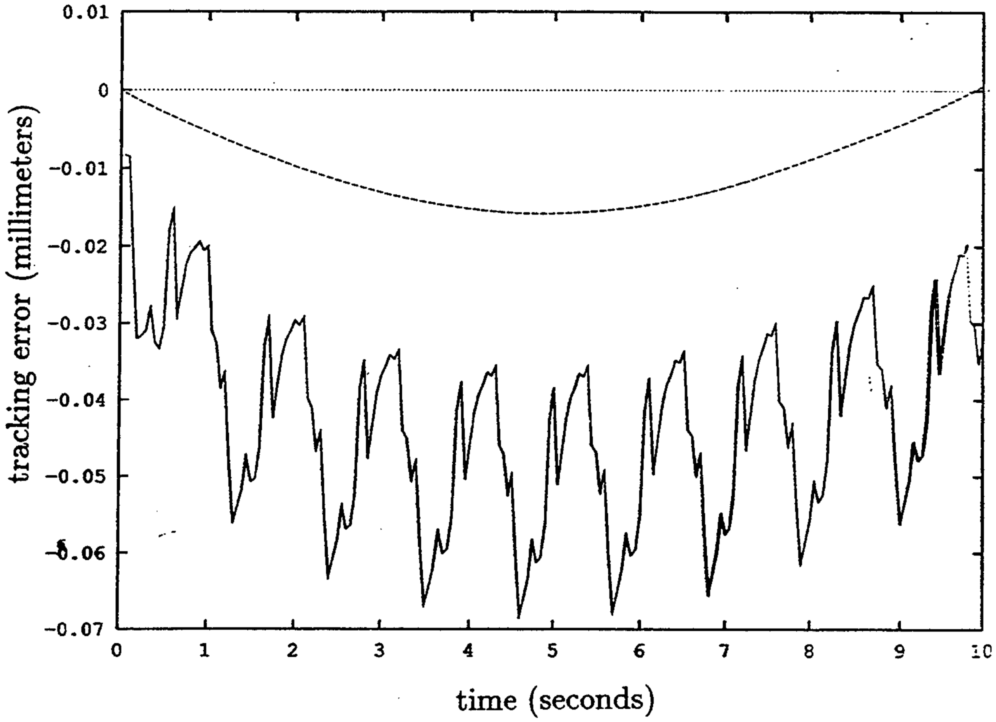


Figure 2: Tracking errors for proposed controller (dashed) and state-feedback controller (solid) in machine tool contouring simulation study

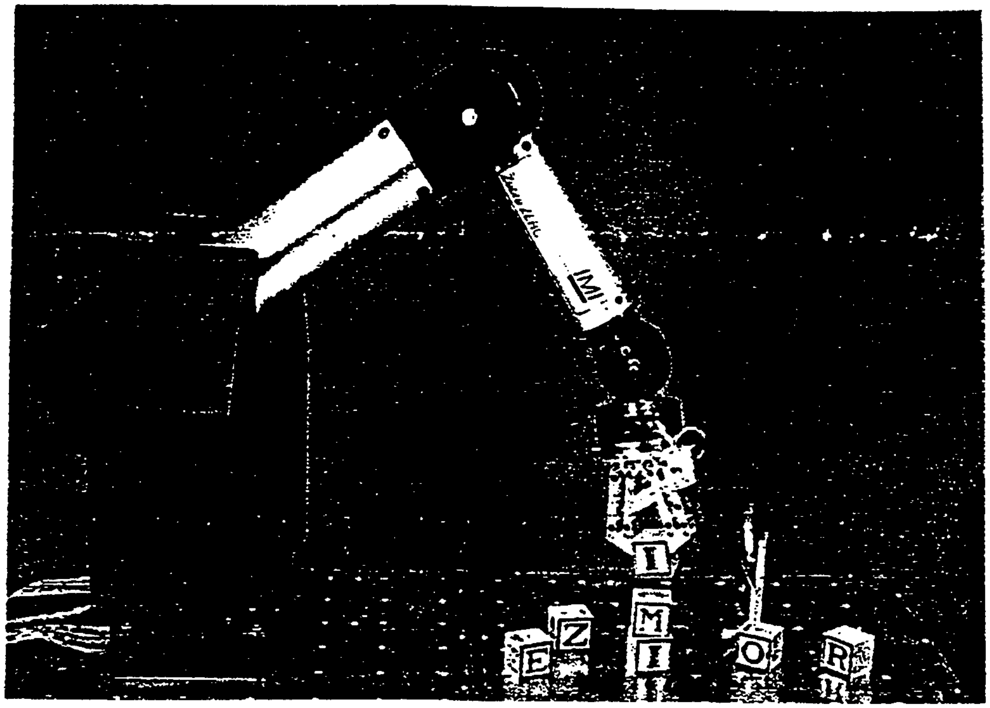


Figure 3: Zebra Robot Manipulator

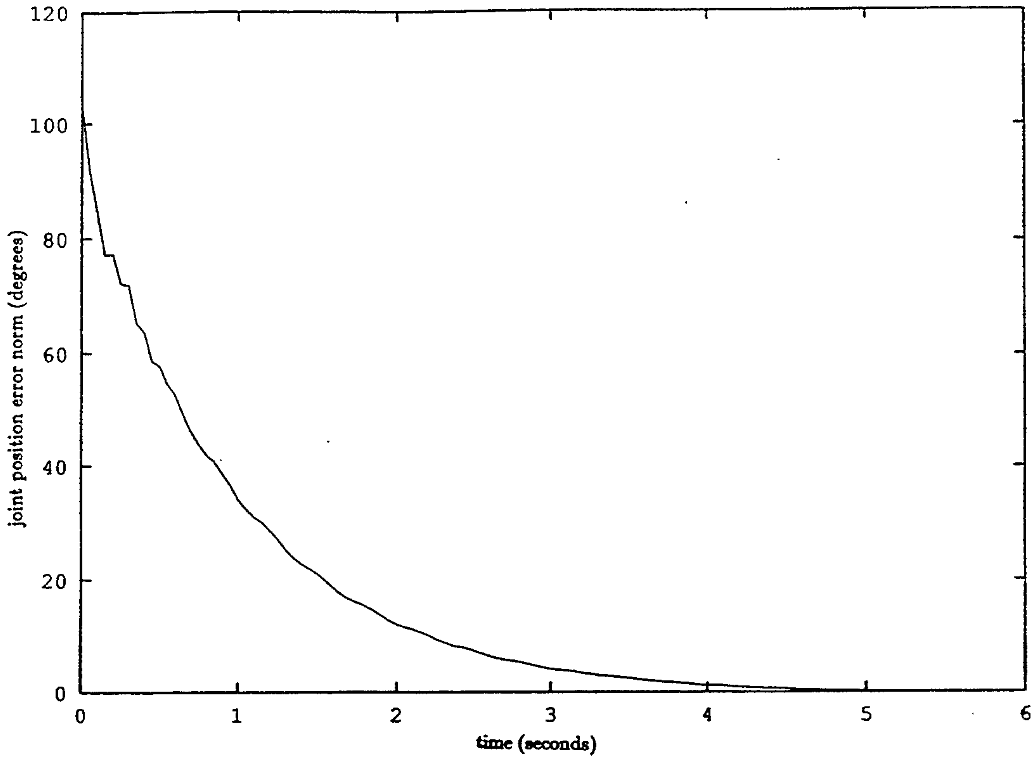


Figure 4a: Response of Zebra robot joint position error norm (proximal three joints) in experimental study

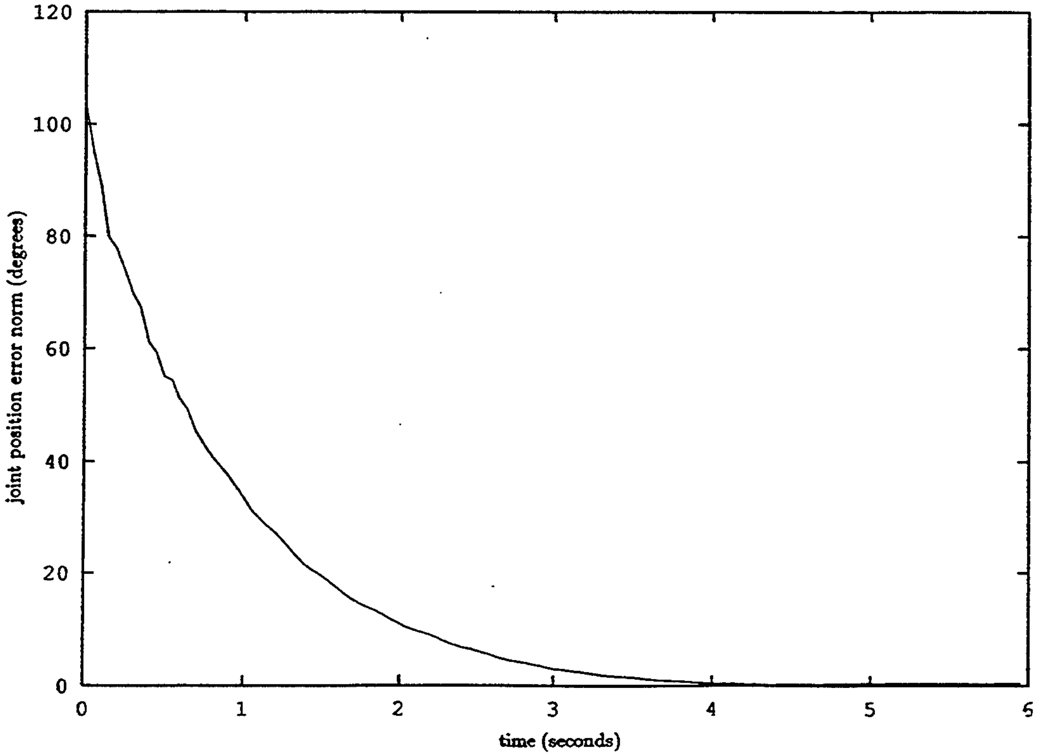


Figure 4b: Response of Zebra robot joint position error norm (distal three joints) in experimental study

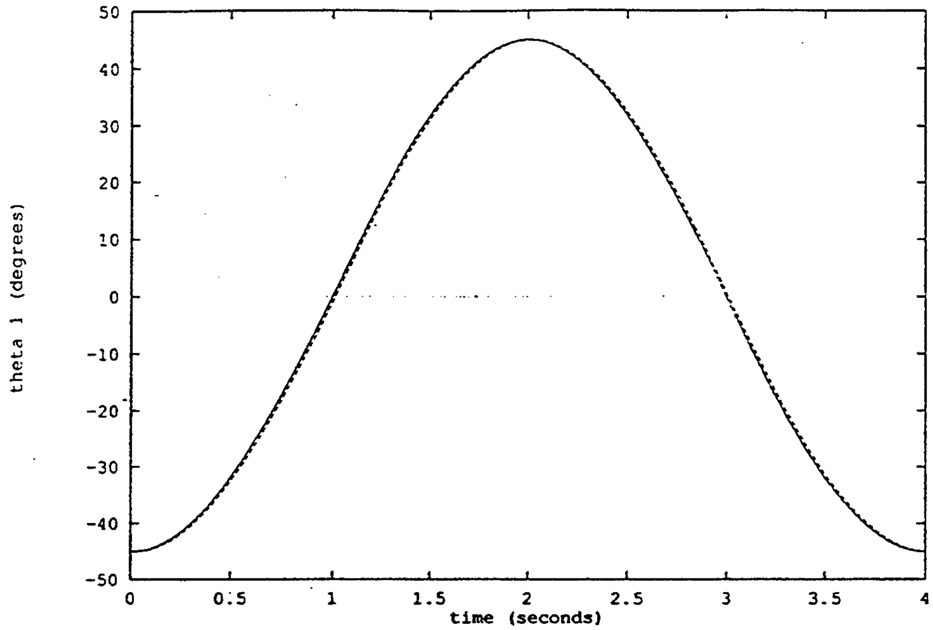


Figure 5a: Desired (dashed) and actual (solid) response of Zebra waist angle in experimental study

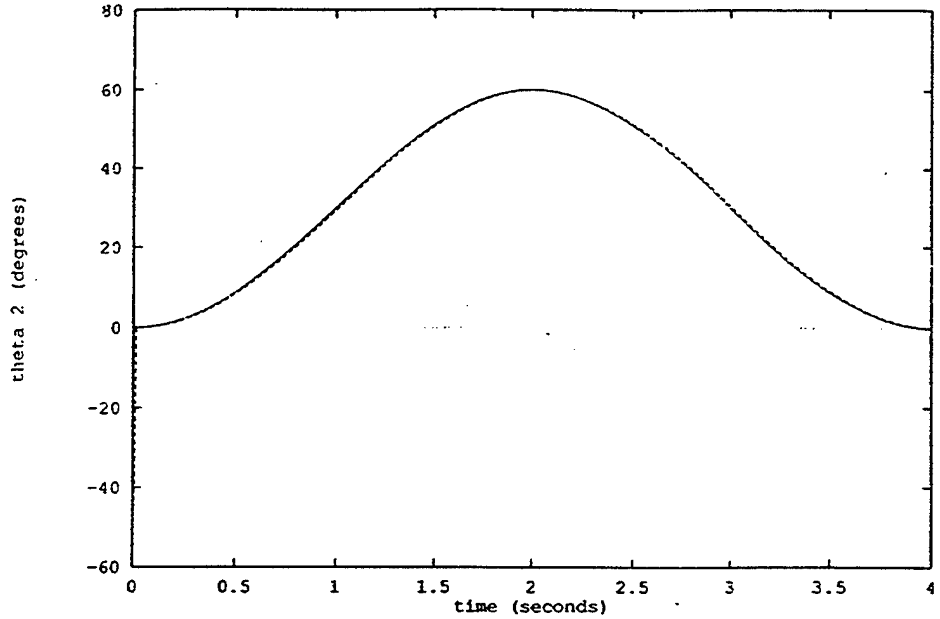


Figure 5b: Desired (dashed) and actual (solid) response of Zebra shoulder angle in experimental study

submitted for publication

NEURAL NETWORK CONTROL OF THERMAL EXPANSION

S. Flachs

and

R. Colbaugh

New Mexico State University
Department of Mechanical Engineering
Box 30001, Dept. 3450
Las Cruces, New Mexico

November 6, 1995

ABSTRACT

In high-precision machining, error involving thermal expansion of the machine body is a major concern. In order to counteract the displacement of the machining tool due to thermal expansion of the machine body, a relationship between the measured temperature distribution of the machine and the displacement of the tool must be determined. One possible solution to this problem is to use a neural network to estimate the displacement of the tool in real-time given the measured temperature distribution. A feasibility study was performed using ANSYS to generate three time-variant temperature distributions of the machine body given three different thermal loadings. ANSYS was also used to determine the resulting machine tool displacements. A backpropagation neural network was then used to estimate the displacement of the machine tool given each of the temperature distributions generated by ANSYS. Finally, a larger neural network was used to encompass the displacements due to all three thermal loading situations. Evaluation of the performance of each neural network was based on a comparison between the tool displacement results of ANSYS and the neural network. Due to small errors between the neural network estimations and the "actual" tool displacement results from ANSYS, it was determined that neural network control of thermal expansion of a machine body is feasible and warrants further study.

INTRODUCTION

High-precision machining is an important part in improved quality and performance in manufacturing. In high-precision machining, tolerances are kept low improving part quality, ease of assembly, and enabling the production of more complex, high-performance designs. In addition, time-consuming finishing processes such as grinding and polishing may be reduced or eliminated. However, errors due to backlash, misalignment, wear, and thermal expansion create difficulties in reducing part tolerances to very low levels. Static geometric errors such as assembly and installation misalignments, machine wear, and backlash are relatively easy to compensate for and many CNC controllers have built-in compensation for these effects [2]. However, thermal expansion error, which may account for 40-70% of dimensional workpiece error [2], is much more difficult to handle.

Thermal errors are dynamic errors which warrant real-time compensation methods. These errors are caused by thermal expansion and distortion of machine components due to heat sources such as motors, bearings, hydraulic systems, increasing room temperature, and sunshine. Thermal expansion and distortion of the machine body can account for positioning errors on the order of 100 micrometers [2].

For real-time compensation of the thermal expansion of the machine body, a relationship between the measured temperature distribution of the machine body and the displacement of the tool location must be determined. Due to the highly non-linear behavior of thermal error, a prediction of tool displacement using traditional heat transfer analysis is extremely difficult. Therefore, other methods of solution must be examined. Non-linear adaptability and fast execution time make neural networks a good candidate for the solution to this problem.

The objective of this study is to asses the feasibility of using neural networks to estimate tool displacement due to thermal expansion of a manufacturing machine body. This estimation may then be used for real-time compensation of this kind of thermal error. For simplicity, the machine body is modeled as a simple two-dimensional solid which is subjected to three types of thermal loadings. A finite element package, ANSYS, is used to generate three time-variant temperature distributions and the resulting tool displacement trajectory for each thermal load case. Three similar neural networks were then developed to estimate the machine tool displacement given a set of discrete temperature values from the three different temperature profiles. The performance of each neural network was evaluated separately based on a comparison between the neural network estimation and the tool displacement results from ANSYS. A comparison is then made to determine which case of thermal expansion is most difficult for a neural network to estimate. Finally, the three thermal loading schemes are combined and a single neural network is developed to estimate the tool displacement due to any of the thermal loading schemes. From the performance of this fairly general neural network, the feasibility of using neural networks to estimate tool displacement due to thermal expansion of a machine body is assessed.

PROBLEM FORMULATION

The first step in the approach to this feasibility study is to model the problem in a simplified but meaningful manner. This involves defining a simplified geometry of the manufacturing machine body, defining thermal and structural boundary conditions, and defining three simple but fairly realistic thermal loading schemes. See Figure #1 for a schematic of the machine model used.

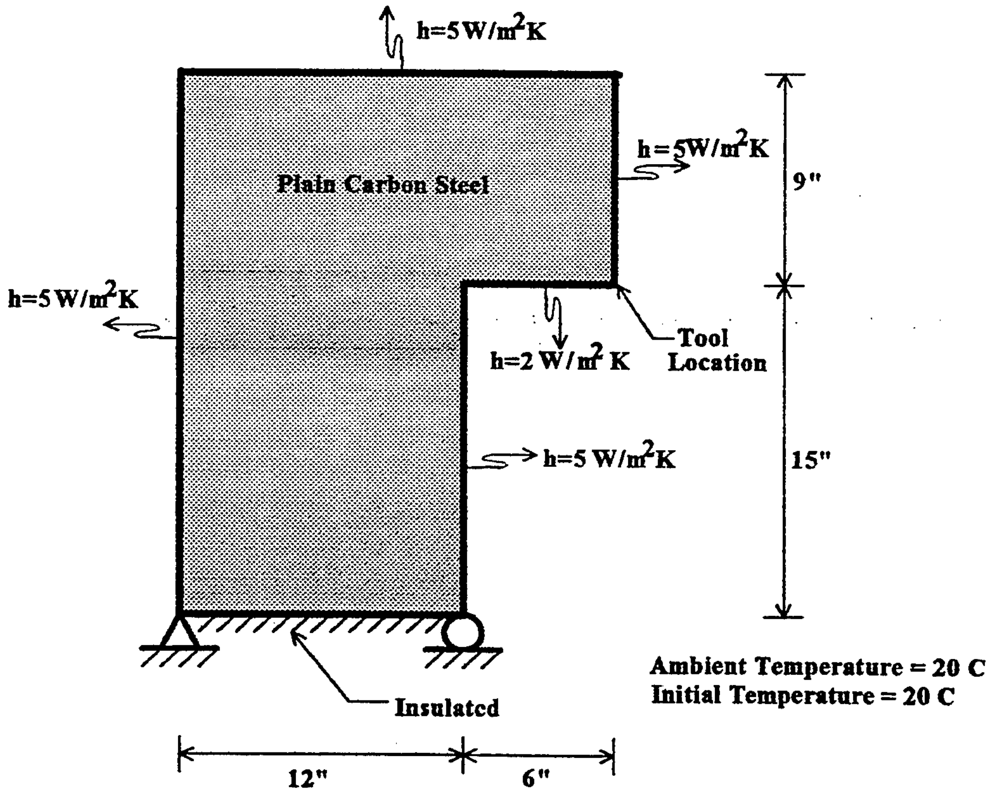


FIGURE # 1: Schematic of Machine Body

Geometry and Composition

The machine body is modeled as a two-dimensional solid shaped like a C-shaped machine frame without the cutting and positioning table. Refer to Figure #1 for dimensions of machine model and location of machine tool. It is assumed the machine is made of plain carbon steel with the following properties:

Density = 7.854 x 10^{-3} kg/m^3 [3],
Specific heat = 434 J/kgK [3],

Thermal Conductivity = 60.5 W/mK [3],
Modulus of elasticity = 200 GPa [1],
Poisson's ratio = 0.29 [1], and
Thermal expansion coefficient = 11.7 x 10^{-6} /°C [1].

Boundary Conditions

Structurally, the machine is modeled to be supported by a pin and roller connection on the floor. At the floor, the machine is also assumed to be insulated. The ambient or room temperature is assumed to be at a constant temperature of 20° C. Initially, the system is assumed to be at steady state with the machine at an initial temperature of 20° C. Also it is assumed there are no external forces acting on the system other than reactions at the floor.

Determination of Convection Coefficients

The machine is assumed to be exposed to convection on all sides except the floor surface. The convection coefficients at these surfaces are estimated using empirical correlations [3]. See Appendix A for calculations. The top surface and all vertical surfaces are associated with a convection coefficient of 5 W/m²K. The bottom exposed surface, however, is associated with a smaller convection coefficient of 2 W/m²K.

Thermal Load Cases

The following thermal load cases are simple but each represent a fairly realistic thermal loading of a manufacturing machine. Figure # 2 shows the schematic of each thermal loading used in this study.

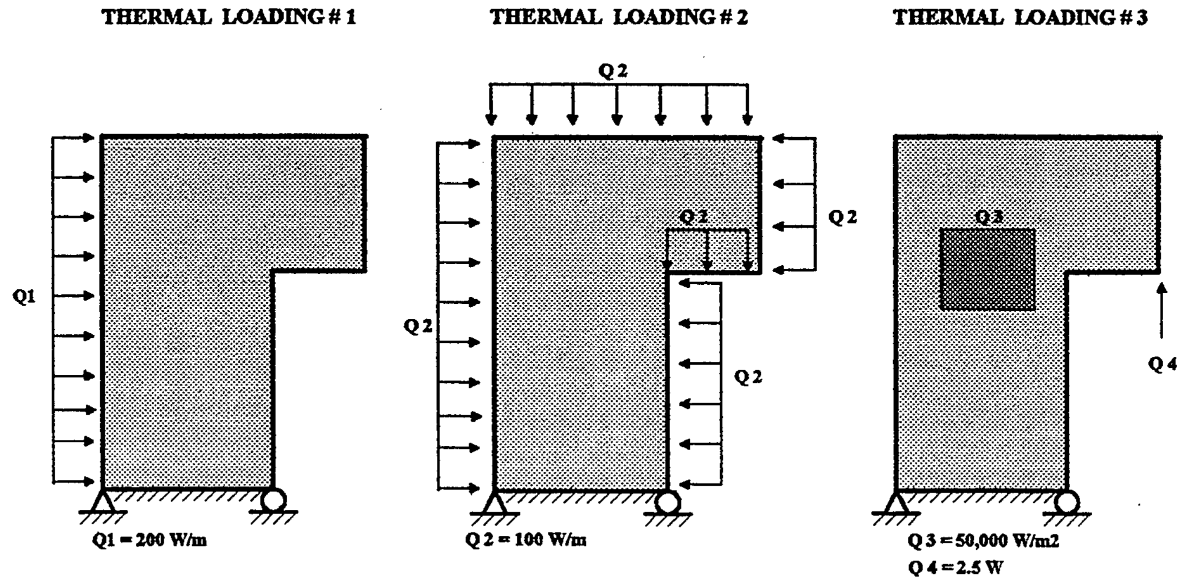

THERMAL LOADING # 1 THERMAL LOADING # 2 THERMAL LOADING # 3

FIGURE # 2: Thermal Load Case Schematics

The first thermal load is the heating of one side of the machine due to sunlight or heat sources such a motor or hydraulic system. The second thermal load is the uniform heating of all sides which may be due to an effect such as increasing room temperature. The third thermal load scheme includes internal heat generation due to a heat source such as an internal motor and a point heat source at the tool location due to heating of the tool during cutting. The heat source, heat flux, and heat generation values used in this study are strictly estimations and do not necessarily represent actual values.

TEMPERATURE PROFILES AND ACTUAL TOOL DISPLACEMENTS

The finite element software package, ANSYS, is used to take the thermal loading and boundary conditions to calculate the time-variant temperature profile for each of the thermal load cases. After this is completed, it is possible for ANSYS to use these temperature profile results and determine the displacement trajectory of the tool location. The displacement trajectories generated by ANSYS are considered to be actual machine tool displacements. In, addition, "measurements" of machine temperature are taken to be discrete temperature values from the time-variant temperature profiles at selected locations on the outer surfaces of the machine. The finite element model consists of 342 elements and 385 nodes. All nodes are uniformly spaced 1 inch apart so that the size of each element is 1 inch x 1 inch. Appendix B contains a complete ANSYS program file used to calculate the time-variant temperature profile and resulting displacement trajectory of the tool location for the first thermal load case. Appendix C and Appendix D contain the changes needed for the second and third thermal loadings, respectively.

Generation of Time-Variant Temperature Profiles

To generate a time-variant temperature profile in ANSYS a transient thermal analysis must be performed which uses the defined thermal loading, boundary conditions, initial condition, convection coefficients, and material properties. The total time span analyzed is 2 hours which must be broken into small increments for solution accuracy. At every time increment, a temperature distribution of the model is calculated. It is expected that most dynamic effects occur when a body is first subjected to a thermal load. Therefore, solution increments should be smaller at first. See Table #1 for the time increments used in this analysis.

Time Interval	Increment Size
0-10 min	30 sec
10-20 min	1 min
20-40 min	2 min
40-120 min	4 min

TABLE # 1: Thermal Solution Increment Size

After the thermal solution is complete, an entire temperature profile of the model is determined at each solution increment. From this data, a temperature history of selected nodes can be extracted to be used as temperature "measurement" inputs to the neural networks.

Determination of Actual Tool Displacements

After the thermal solution is completed, ANSYS can use the temperature profile results, structural boundary conditions, and material properties to determine the displacement trajectory of the tool location. To do this, a static structural analysis is performed to compute the thermal expansion of the finite element model at desired time steps. ANSYS is then used to determine the x and y displacement components of the model at the tool location. This information is used to generate the displacement trajectory of the machine tool. In this analysis, the x and y displacement components of the tool location are computed every 2 minutes for the first 40 minutes and every 4 minutes thereafter.

DEVELOPMENT OF NEURAL NETWORKS

The neural networks developed in this study use temperature "measurements" generated by ANSYS as inputs and output estimations of the x and y components of the tool displacement. See Figure #3 for a functional representation of the neural networks.

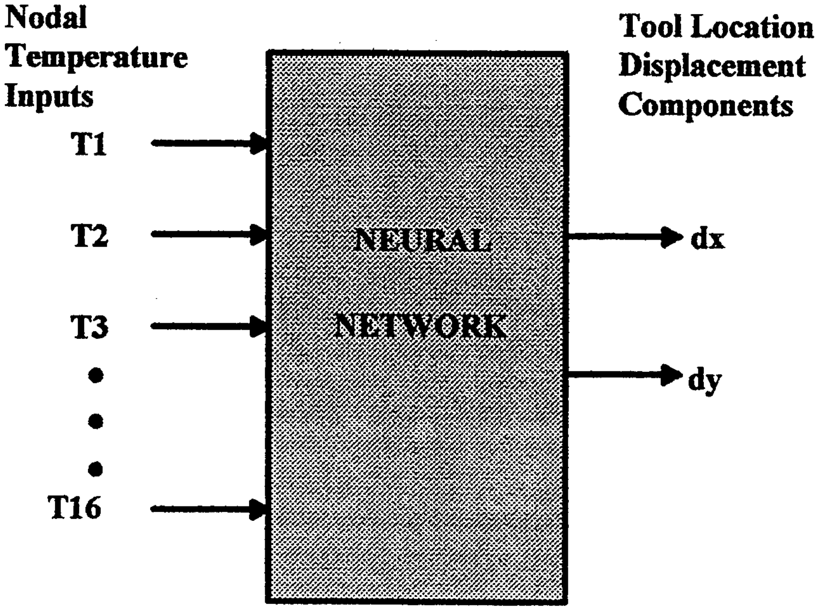


FIGURE # 3: Black-box Representation of Neural Network

A modified backpropagation technique is used to train the neural networks so that the output estimation of tool displacement is sufficiently close to the "actual" values generated by ANSYS. If a neural network is unable to sufficiently estimate the tool displacement after a certain amount training, the network structure is modified and training is repeated. Evaluation of each neural network performance is based on a comparison between the tool displacement results of ANSYS and the neural network.

Data Preprocessing

From the thermal solution, a temperature history of selected nodes can be extracted by ANSYS to be used as temperature "measurement" inputs to the neural networks. Therefore, it must be determined which nodes to use for these temperature "measurements". Nodes on the exterior surfaces of the model are a practical choice because here is where it would be easiest to mount temperature measuring devices such thermocouples. In addition, it was found that thermal displacements in the x-direction were larger than in the y-direction. So more nodes are chosen on the horizontal surfaces than on the vertical surfaces. Figure #4 shows which nodes are for the selected temperature "measurement" locations. There are a total of 16 nodes selected.

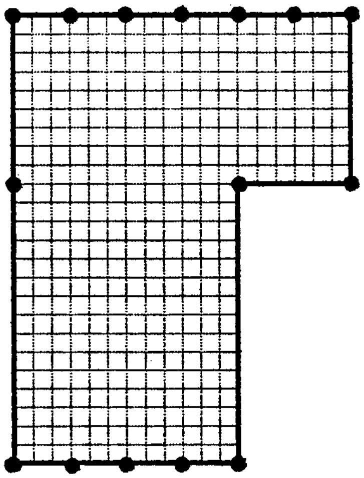


FIGURE # 4: Nodes Selected For Temperature "Measurements"

The temperature "measurements" must be taken at a selected sampling period. For the temperature data to correspond with the tool trajectory data, a "measurement" is taken every 2 minutes for the first 40 minutes and every 4 minutes thereafter. This sampling method was found to sufficiently represent the dynamics of all types of thermal expansion.

Now, the temperature and displacement data must be normalized before it can be used with a neural network. The type of neural network used in this study accepts inputs in the range of -1.0 to +1.0 and during training, the desired or actual outputs must be in the range of 0.0 to +1.0. Therefore, to avoid errors associated with neural network operation at the endpoints of these ranges, the temperature data is normalized to be between -0.9 and +0.9 and the displacement data is normalized to be between +0.1 and +0.9. Equations 1 and 2 are used to normalize the temperature and displacement data, respectively.

$$T_n = 0.9 * \frac{T - (T_{\min} + \frac{T_{\max} \ T_{\min}}{2})}{\frac{T_{\max} - T_{\min}}{2}} \qquad (1).$$

$$dx_n = 0.5 + 0.8 * \frac{dx - (dx_{\min} + \frac{dx_{\max} - dx_{\min}}{2})}{dx_{\max} - dx_{\min}} \qquad \text{and}$$

$$dy_n = 0.5 + 0.8 * \frac{dy - (dy_{\min} + \frac{dy_{\max} - dy_{\min}}{2}}{dy_{\max} - dy_{\min}} \qquad (2).$$

Where: T, dx, and dy are the temperature and displacement data,
T_n, dx_n, and dy_n are the normalized temperature and displacement data,
T_{min}, dx_{min}, and dy_{min} are minimum temperature and displacements, and
T_{max}, dx_{max}, and dy_{max} are maximum temperature and displacements.

The neural network training sets consist of the 16 normalized temperature "measurement" input features and the 2 normalized desired output components. Each of the 16 input features and the 2 desired output vectors used in the training set consist of every other data point of the entire time span. In other words, a data point is used for the training set every 4 minutes for the first 40 minutes and every 8 minutes thereafter.

To test the performance of each trained neural network, a testing set is made up of the 16 input features using all data points of the entire time span. Therefore during testing, the neural networks essentially interpolate between training data points.

In combining the three thermal loading schemes, the following table summarizes the approach taken for all input and desired output features. As before, every data point is used in the testing set and every other data point is used in the training set.

Time	Thermal Load Case #
0	1
0	2
0	3
1	1
1	2
1	3
...	...

TABLE # 2: Combination of Three Thermal Load Cases

Neural Network Training Scheme

The neural networks used in this study have the basic structure shown in Figure #5. The structure is a three layer, fully inter-connected structure where each connection is associated with a weight value. The input layer consists of 16 nodes corresponding to "measurements" of the machine temperature distribution. The output layer consists of 2 nodes corresponding to the displacement components of the machine tool. Between the input and output layers is a hidden layer which performs feature extraction and noise suppression [2].

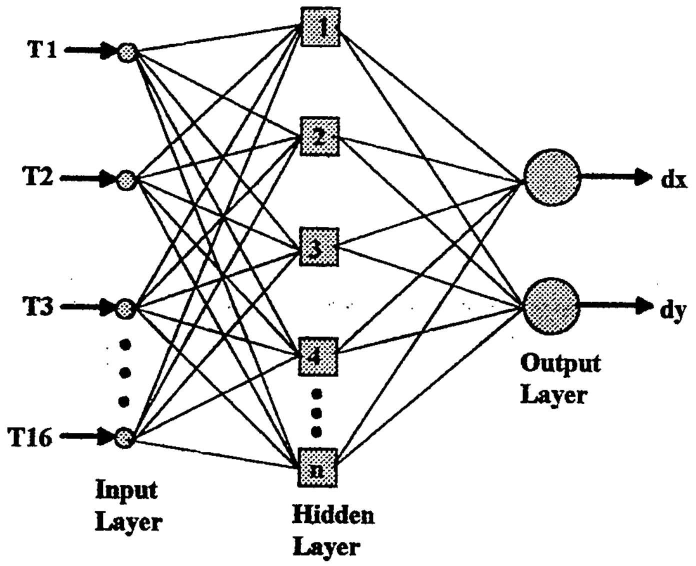


FIGURE # 5: **Neural Network Structure**

The first step in training the neural networks is gathering and preprocessing sufficient input and output data. Then the learning parameters, or weights, of the neural network are initialized randomly. Now, the input temperature data is applied to the network and the outputs are compared to the known tool displacement outputs. The weights are then adjusted by a backpropagation rule which essentially adjusts the weights in the direction of the steepest error gradient. In normal backpropagation, the weights are adjusted in this manner until the system output error reaches a desired level. The system output error is defined as

$$E = \frac{1}{4n} \sum_{1}^{n} (|e_1| + |e_2|)^2 \qquad (3).$$

Where: E = system output error,
n = number of training samples,
e_1 = error between known and estimated x-component of tool displacement, and
e_2 = error between known and estimated y-component of tool displacement.

However, with normal backpropagation, neural network training might lead to a relative or local error minimum instead of the absolute or global minimum. The modified technique used in this study, is intended to eliminate this problem. As in normal backpropagation, weights of the neural network are initialized randomly, the input temperature data is applied to the network and the outputs are compared to the known tool displacement outputs. The weights are adjusted by a backpropagation rule for 100 iterations. The final weights, system error, and convergence rate are saved. The weights are then reinitialized and adjusted for 100 more iterations. This process is repeated 100 times. When finished, the training scheme uses the weights from the best network and continues with normal backpropagation until the desired error level is reached. The best solution is determined by a linear combination of the final system error and the error convergence rate.

Testing and Evaluation of Neural Networks

Trained neural networks are tested by applying the testing set which consists of the 16 input temperature "measurements" over the entire time span. Evaluation of individual neural network performance is based on the percentage error of estimation to actual tool displacement for both x and y directions. A comparison is made to determine which case of thermal expansion is most difficult for a neural network to estimate. This is done based on the system output errors and the time-dependant percentage errors. For comparison validity, neural networks of all thermal expansion cases were trained for the same amount of time.

ANALYSIS OF FIRST THERMAL LOADING

Time-Varient Temperature Profile

The temperature profile generated by ANSYS of the machine body with a constant heat flux applied to one side of the machine is shown in Figure #6. This profile is generated at the end of the time span, 2 hours. It can be seen, that the temperature distribution is nearly constant in the vertical direction, this implies most thermal expansion will occur in the horizontal direction. It should also be noted that the maximum temperature change occurring in the machine body is less than 2°C and the temperature gradient from the left-hand side to the right-hand side is less than 1°C.

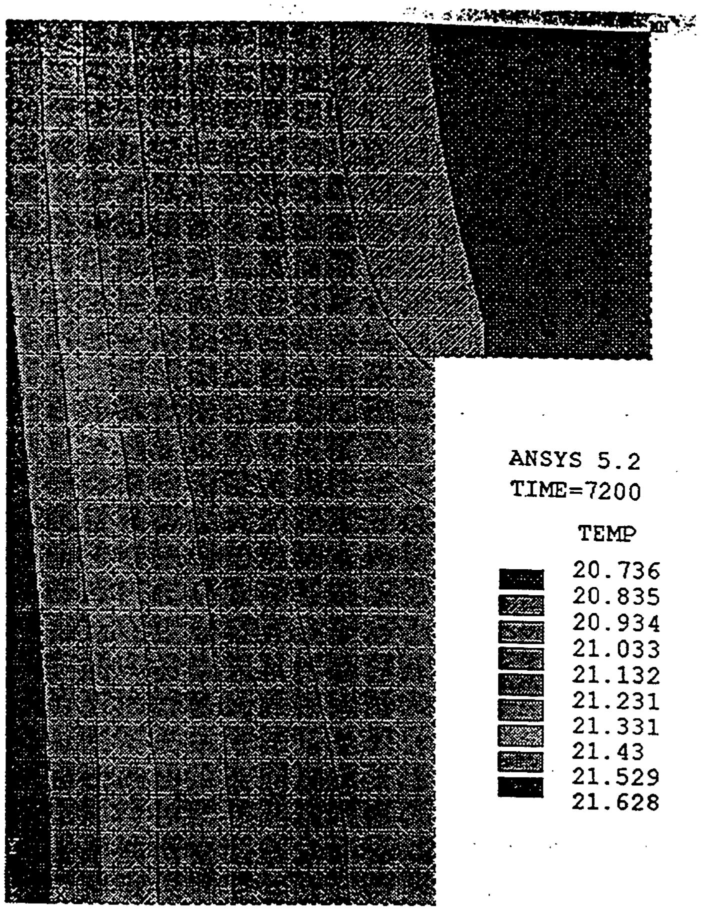


FIGURE # 6: Final Temperature Profile (Thermal Load #1)

Actual Tool Location Displacement

The tool displacement results generated by ANSYS for a uniform heat flux applied to the left side of the machine are shown in Figure #7. The tool moves approximately 7μm in the x-direction and 3μm in the y-direction. Also, there is a hinge effect that takes place in the first 20 minutes, then the displacement becomes linear.

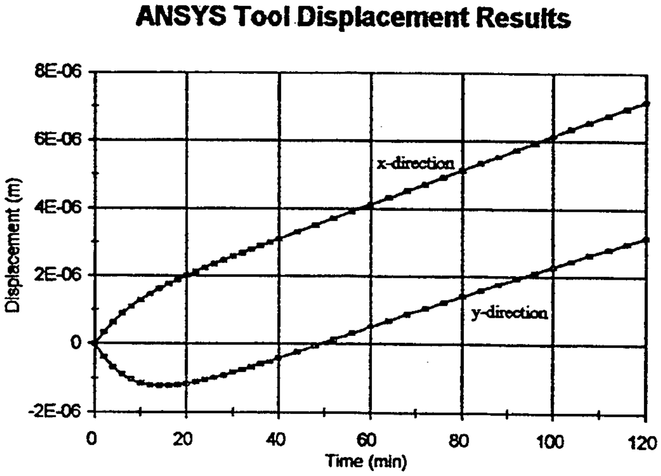


FIGURE # 7: Actual Tool Displacement (Thermal Load #1)

Performance of Neural Network

A 3 layer neural network with 10 nodes in the hidden layer was trained to produce the error shown in Figure #8. The estimation results for this thermal loading are within 3% of the tool displacement determined by ANSYS. The percentage errors are larger in the early stages of expansion which may be due to extremely small temperature gradients. These small gradients are more susceptible to errors such as round-off errors. However, after approximately 30 minutes the percentage error converges to zero .

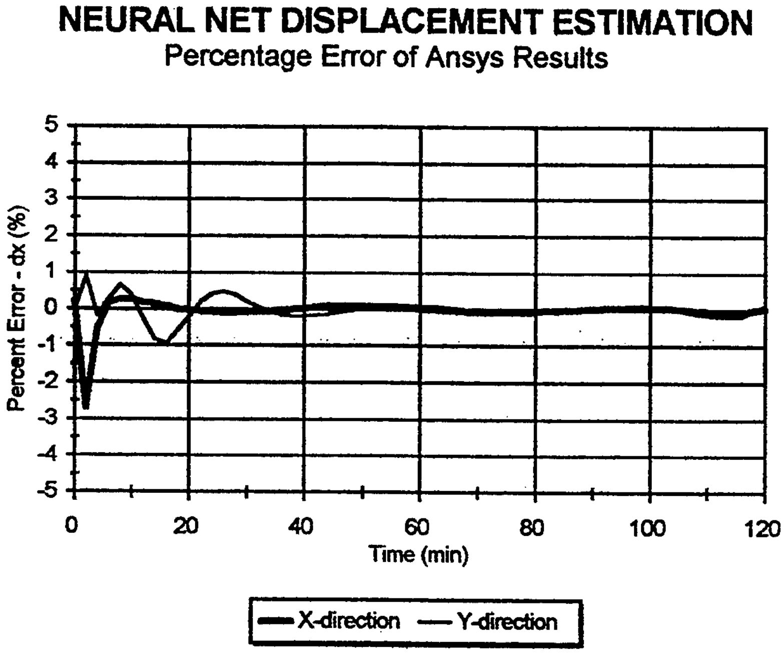


FIGURE # 8: **Neural Network Performance (Thermal Load #1)**

ANALYSIS OF SECOND THERMAL LOADING

Time-Varient Temperature Profile

The temperature profile of the machine body with a constant heat flux applied to all exposed sides of the machine is shown in Figure #9. This profile is generated at the end of the time span, 2 hours. As expected, the machine is coolest at the bottom insulated surface and warmest near the exposed corners. The maximum temperature change occurring in the machine body over the time span just over 2°C and the temperature gradient from warmest to coolest after 2 hours is less than 1°C.

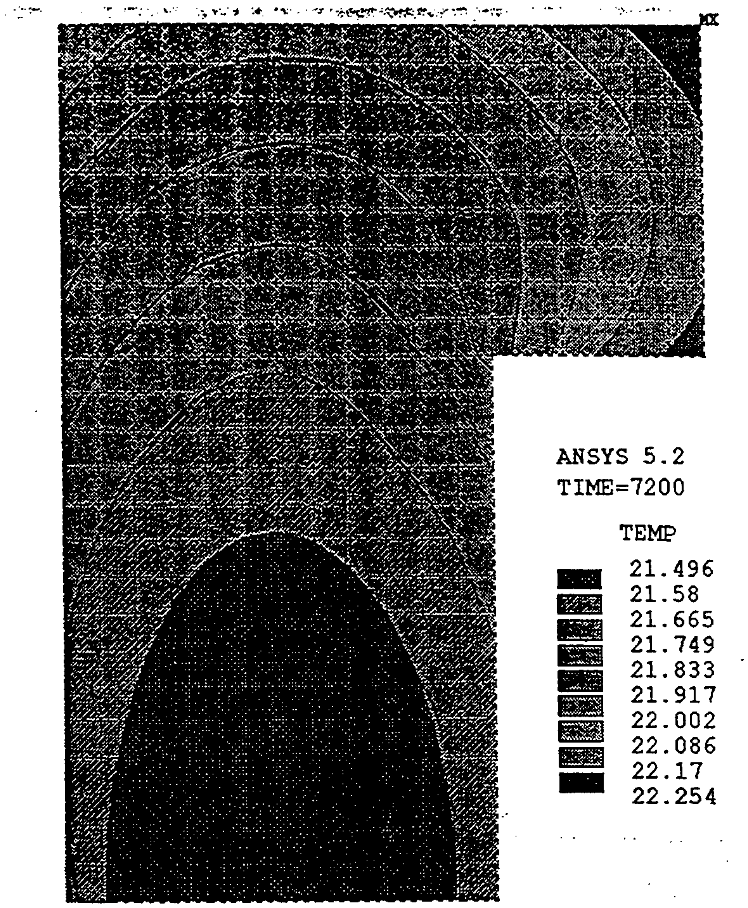


FIGURE # 9: **Final Temperature Profile (Thermal Load #2)**

Actual Tool Location Displacement

The tool displacement results generated by ANSYS for a uniform heat flux applied to all sides of the machine are shown in Figure #10. The tool moves approximately 9.5μm in the x-direction and 6.5μm in the y-direction. Both displacement components become mostly linear after approximately 10 minutes.

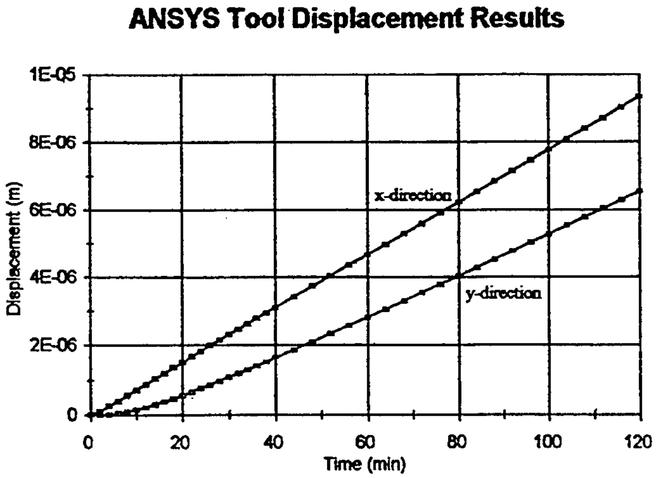


FIGURE # 10: **Actual Tool Displacement (Thermal Load #2)**

Performance of Neural Network

A 3 layer neural network with 10 nodes in the hidden layer was trained to produce the error for the second thermal loading shown in Figure #11. This neural network is similar to the network trained for the first thermal loading except the weights are different. The estimation results for this thermal loading are within 5% of the tool displacement determined by ANSYS. The percentage errors are again larger in the early stages of expansion which may be due round-off errors. After approximately 40 minutes the percentage error converges nicely to zero.

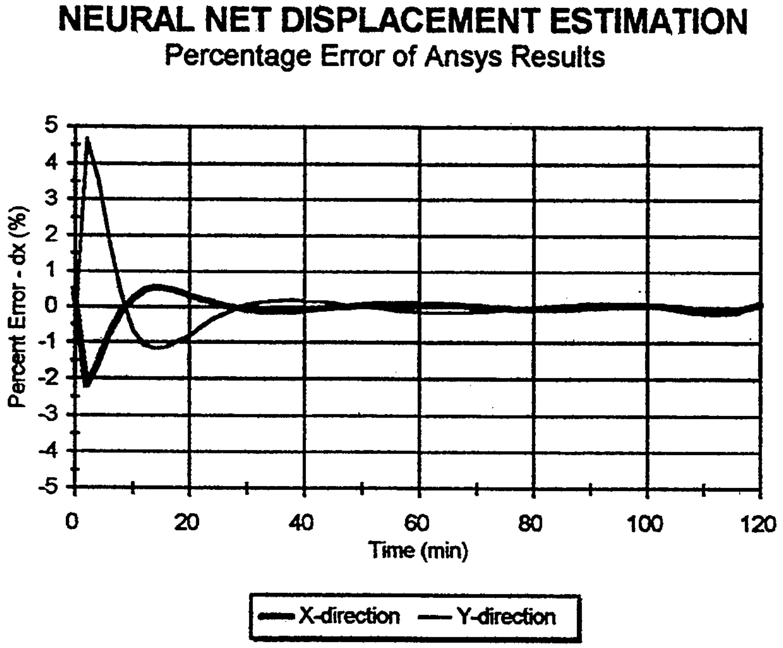


FIGURE # 11: Neural Network Performance (Thermal Load #2)

ANALYSIS OF THIRD THERMAL LOADING

Time-Varient Temperature Profile

The machine body temperature profile resulting from heat generation in the center of the machine and a point heat source due to tool heating is shown in Figure #12. This profile shows the state of the machine at the end of 2 hours. As expected, the machine is coolest at the bottom insulated surface and warmest in the center near the heat generation source. The point heat source applied at the tool location does not seem to affect the machine body. However, it is possible that the strength of the heat source used is not realistic. The maximum temperature change occurring in the machine body over the time span less than 3°C and the temperature gradient from warmest to coolest after 2 hours is less than 1.5°C.

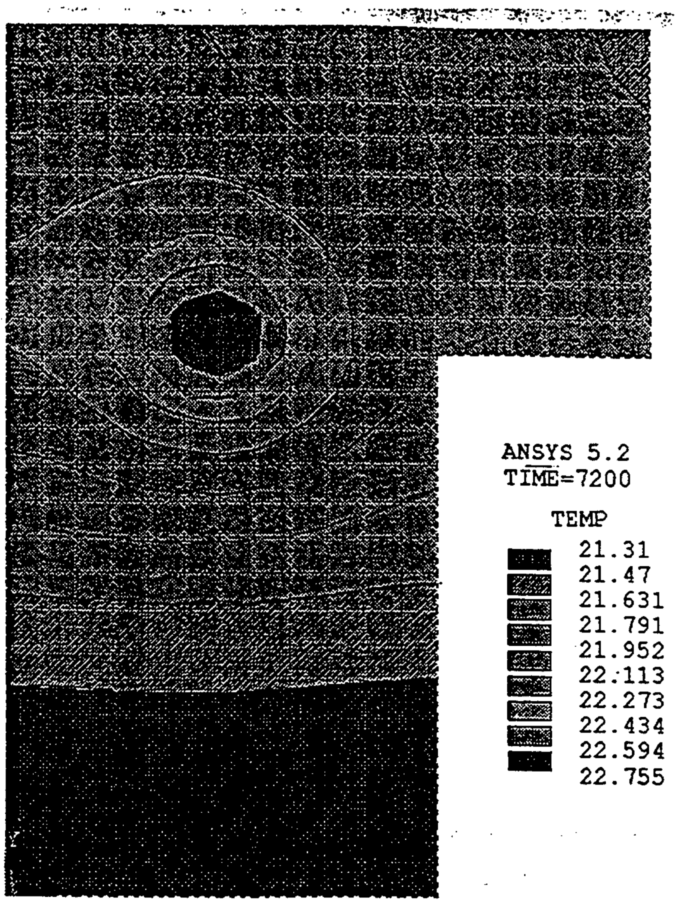


FIGURE # 12: **Final Temperature Profile (Thermal Load #3)**

Actual Tool Location Displacement

Figure #13 shows the tool displacement trajectory generated by ANSYS for the third thermal loading. It can be seen that the tool moves approximately 9.5μm in the x-direction and 8μm in the y-direction. Again, both displacement components become mostly linear after a short period of time, approximately 10 minutes.

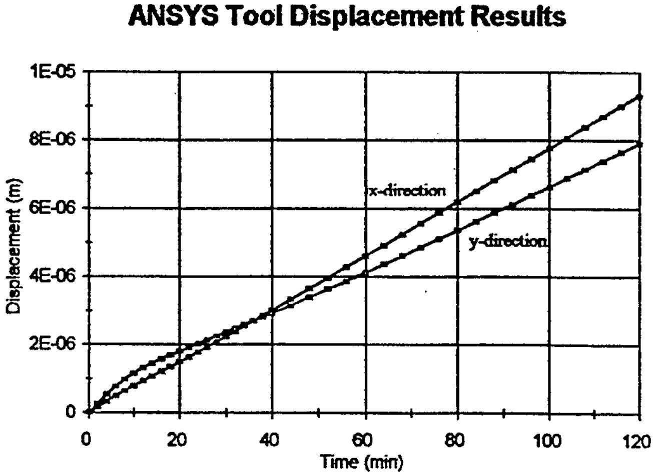


FIGURE # 13: **Actual Tool Displacement (Thermal Load #3)**

Performance of Neural Network

The neural network displacement estimation performance for the third thermal loading is shown in Figure #14. The neural network used consists of 3 layers with 10 nodes in the hidden laye, similar to the networks trained for the first and second thermal loadings. Figure #14 shows the estimation results for this thermal loading are within 3% of the tool displacement determined by ANSYS. This percentage error is a bit larger during the first hour than those of the first and second thermal loadings, indicating the relationship between the temperature "measurements" and the machine tool displacement is not quite as good during this time. Possibly a longer training time for the neural network is needed to better fit the features of this relationship.

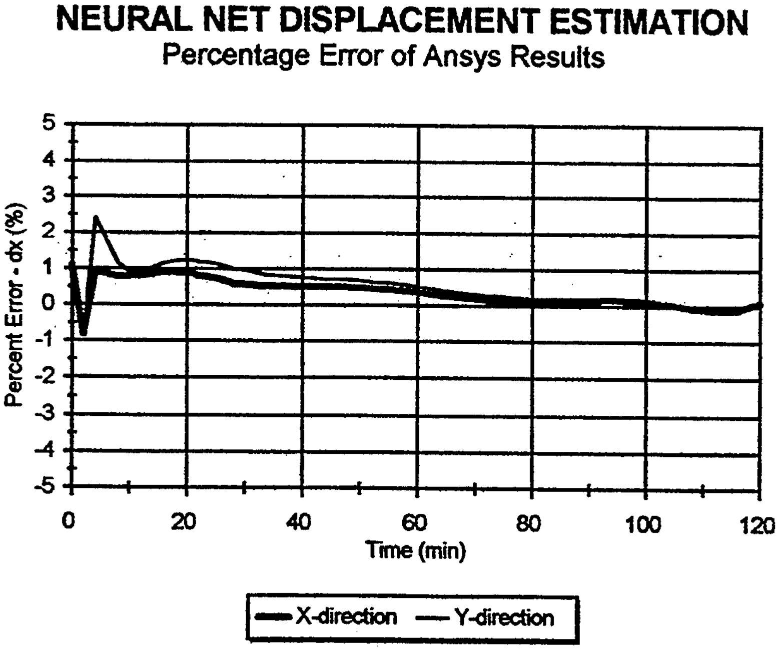


FIGURE # 14: Neural Network Performance (Thermal Load #3)

ANALYSIS OF COMBINED THERMAL LOADING

Performance of Neural Network

The performance of the neural network encompassing all three thermal load cases is shown in Figure #15. The percentage error of the estimated displacement to the displacement generated by ANSYS is well under 1.5% for the entire time span. The method of combining the three sets of data is probably responsible for the high frequency oscillation seen in the percentage error plot. A 3 layer neural network with 20 nodes in the hidden layer is used to generate these results. It is unexpected that such a small network will perform so well for a fairly general set of conditions.

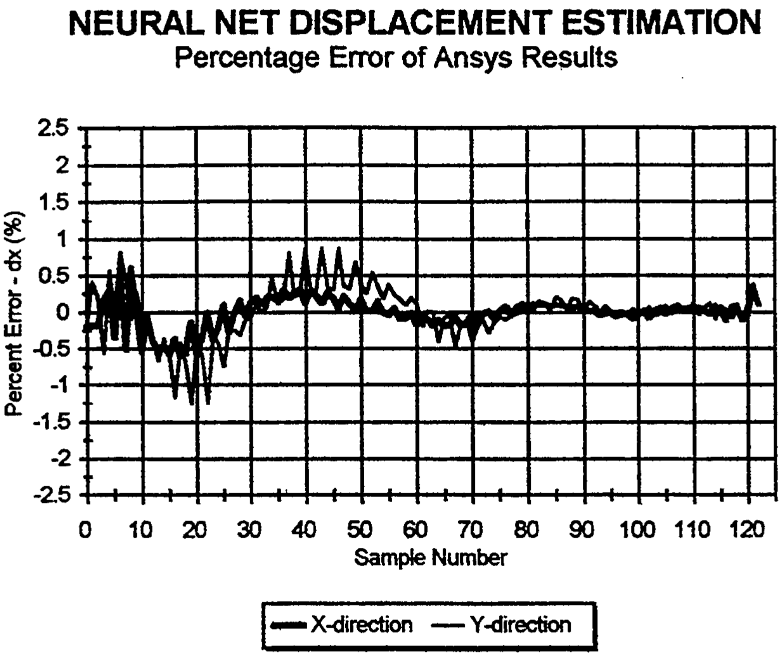


FIGURE # 15: Neural Network Performance (Combined Thermal Load)

DISCUSSION OF RESULTS

The machine body temperature gradients generated in this study are less half of the 4°C gradient found by Chen in his examination of thermal expansion of a CNC machine [2]. Chen's experiment was run for 7 hours while this study examines a 2 hour time span. This implies the thermal expansions used in this study are reasonable estimations of what may be expected in actual applications.

Neural network performance for all three thermal load cases and the combined thermal load case is summarized in Table #3. It can be seen that the neural network structure used trained

Thermal Load Case	Neural Network Structure	Training Time (iterations)	System Output Error	Max. Percent Error (%)
# 1	1 hidden layer with 10 nodes	60,000	0.000011	2.684
# 2	1 hidden layer with 10 nodes	60,000	0.000043	4.661
# 3	1 hidden layer with 10 nodes	60,000	0.000031	2.415
Combined	1 hidden layer with 20 nodes	60,000	0.000042	1.235

TABLE # 3: Comparison of Neural Network Performance

better on the first and third thermal loadings than the second. This may be due to the fact that there is a total of 5 separate loads applied to the perimeter of the machine. The first thermal load case only has 1 load applied and the second load case has 2 loads applied. The system output error of the first thermal loading is significantly lower than the third thermal loading due to the fact the error did not converge to zero as quickly in the third thermal load case. This effect is probably due to the fact that the temperature "measurements" were all taken on the perimeter of the machine body and could not represent the internal heat generation expansion as efficiently as the expansion due to a heat flux applied to an outside surface. This theory is corroborated by the fast convergence of the second load case which also has only thermal loads applied to the exterior of the machine.

The comprehensive neural network developed in this study performed exceptionally well. It has only double the nodes of the individual neural networks with an maximum percentage error of only 1.2%. The method of combining the three sets of data is probably responsible for small high frequency oscillations in the percentage error. As a result, the system output error is a bit higher than those of the individual thermal load neural networks. All neural networks exhibit a low frequency oscillatory nature in the percentage error. It is not known at this time what causes this effect.

It should also be noted that all networks used in this study are fairly small neural networks. As a rule of thumb, the number of nodes in the hidden layer should at least double the number of inputs to the neural network. Then for this system, a reasonable size network consists of approximately 32 nodes in the hidden layer. Each of these networks, even the comprehensive network is significantly smaller than this rule of thumb. This would imply there is quite a bit of room to add complexity to the problem without generating an extremely large neural network.

CONCLUSIONS

Due to the exceptional performance exhibited by the comprehensive neural network, it is determined that neural network estimation of thermal expansion of a machine body is feasible at this level of complexity. In addition, it was found that a relatively small neural network can estimate the thermal displacement of the machine tool with less than a 1.5% error. Of the three types of thermal loading investigated, thermal expansion due to a constant heat flux applied to all exposed surfaces was the most difficult for a neural network to estimate. This is probably due to the fact that this type of loading consists of the most individual thermal loads. It was also found that in the early stages of thermal expansion due to internal heat generation, it is slightly more difficult for a neural network to predict tool displacement using temperature "measurements" taken from the exterior of the machine body. Finally, all neural networks used in this study exhibit a low frequency oscillatory nature in percentage error. It is not known at this time what causes this effect.

RECOMMENDATIONS

Further work in this area of study should include an optimization of temperature sensor location for satisfactory neural network estimation. Also, the effects of temperature measurement noise or error should be determined. This could include adding a Gaussian white noise or other type of noise to the temperature results obtained from ANSYS and evaluating the effect on neural network performance.

REFERENCES

[1] Boresi, Arthur P., Schmidt, Richard J., Sidebottom, Omar M., **Advanced Mechanics of Materials**, Fifth Edition, John Wiley & Sons, Inc., 1993.

[2] Chen, Jenq-Shyong, "Computer-Aided Accuracy Enhancement For Multi-Axis CNC Machine Tool", **International Journal of Machine Tools and Manufacture**, Vol. 35, No. 4, April, 1995.

[3] Incropera, Frank P., DeWitt, David P., **Fundamentals of Heat and Mass Transfer,** Third Edition, John Wiley & Sons, Inc., 1990.

[4] Flachs, Gerald, "EE 562 Lecture Notes", New Mexico State University.

[5] Flachs, Sharlene, "Introduction to Neural Networks", Presentation, Delphi Saginaw Steering Systems, August, 1995.

[6] Fu, Li Min, **Neural Networks in Computer Intelligence**, McGraw-Hill, Inc., 1994.

[7] Pao, Yoh-Han, **Adaptive Pattern Recognition and Neural Networks**, Addison-Wesley Publishing Company, Inc., 1989.

[8] Xu, Wei, "Finite Element Grid of Tombstone", ANSYS Command Program, New Mexico State University.

to appear: International Journal of Robotics and Automation

A ROBUST APPROACH TO REAL-TIME TOOL WEAR ESTIMATION

R. Colbaugh K. Glass

Department of Mechanical Engineering
New Mexico State University, Las Cruces, NM 88003

Abstract

This paper presents a robust strategy for estimating tool wear in metal cutting operations. The proposed estimation algorithm consists of two components: a recurrent neural network to model the tool wear dynamics, and a robust observer to estimate the tool wear from this model using measurements of cutting force. It is shown that the algorithm ensures that the tool wear estimation error is uniformly bounded in the presence of bounded unmodeled effects, and that the ultimate bound on this error can be made as small as desired. The proposed approach is applied to the problem of estimating tool wear in turning and is shown to provide wear estimates which are in close agreement with experimental results.

1. Introduction

There is considerable interest in increasing the level of automation in manufacturing processes. Included among the anticipated benefits of this automation are enhanced productivity, improved product quality, and reduced manufacturing costs. However, progress toward the development of automated manufacturing systems has been hindered by the lack of dependable machine tool diagnostic schemes. In particular, the problem of reliably measuring the tool wear has been a significant obstacle to the realization of effective automated machining systems. The approaches proposed thusfar for measuring tool wear can be divided into two broad classes: direct methods and indirect methods. In the direct approach, tool wear is measured directly using, for example, microscopes or computer vision technology. While this method can provide very accurate wear information, it is inherently an off-line approach and is therefore limited in its applicability. Alternatively, the indirect approach to tool wear measurement is to estimate the wear state from measurable cutting variables, such as cutting force or acoustic emission. Observe that the indirect method can be used for real-time tool wear determination and consequently has considerable potential for use in automated machining systems. However, the accuracy of existing indirect techniques is often unsatisfactory.

Recognizing the importance of reliably estimating the machine tool wear state in real-time, several researchers have studied this problem in recent years [e.g., 1-10]. Early work focused on the application of statistical methods to associate patterns in measurable signals with wear states [e.g., 1,2]. More recently, it has been proposed that this pattern recognition could be accomplished effectively using neural networks [3-5]. A different approach to tool wear estimation is to acknowledge that tool wear evolves as a dynamical system during the cutting process, and to apply system theoretic ideas to the estimation of wear state [e.g., 6-10]. The principle difficulty associated with this strategy is the nonlinear nature of the process model and the lack of good estimates for the model parameter values, so that traditional state observation techniques are difficult to apply. Nevertheless, this method has considerable potential because the formulation is compatible with the dynamic character of the tool wear process.

This paper presents a new strategy for the real-time estimation of tool wear in metal cutting operations. The proposed estimation algorithm consists of two components: a recurrent neural network to model the tool wear dynamics, and a robust observer to estimate the tool wear from this model using measurements of cutting force. Thus the approach is to train the neural network so that it provides a good model for the nonlinear and uncertain tool wear dynamical system, and then to design the observer to ensure good performance of the overall system in the presence of (inevitable) modeling error. A stability analysis is conducted to show that tool wear estimation error is uniformly bounded with an ultimate bound that can be reduced arbitrarily. The efficacy of the proposed approach is demonstrated through a case study involving the estimation of tool wear in a turning application; it is shown in this study that the algorithm provides wear estimates which are in close agreement with experimental results.

2. Preliminaries

This paper considers the problem of estimating in real-time the tool wear state during machining operations. Numerous researchers have studied the cutting process and proposed models for tool wear. For example, Park and Ulsoy [9] have proposed that flank wear dominates the tool wear process in many important applications, and have proposed the following model for this component of tool wear:

$$\begin{aligned} \dot{w}_{f_1} &= -k_1 v w_{f1} + k_2 \frac{Fv}{fd} \\ \dot{w}_{f_2} &= k_3 \sqrt{v} \exp(\frac{-k_4}{T}) \\ F &= k_5 f^n d + k_6 vd + k_7 d + k_8 d(w_{f_1} + w_{f_2}) \end{aligned} \tag{1}$$

where w_{f_1} and w_{f_2} are the flank wear components due to abrasion and diffusion, respectively, $w_f = w_{f_1} + w_{f_2}$ is the total flank wear, f, v, d are the feed, cutting speed, and depth of cut, respectively, and are the inputs for the system, and F is the cutting force and is the output (measurable variable) for the system. In (1), n and the k_i are constants which depend on the effective rake angle, workpiece material, and other task-related parameters, and T is the tool-work temperature and depends on the inputs f, v and the state w_{f_1}, w_{f_2}. Observe that the tool wear model (1) is a complicated and highly uncertain dynamical system. For example, the model for tool-work temperature is purely an empirical one, and only crude estimates are available for the values of most of the parameters which appear in (1). Indeed, although the model (1) has shown reasonable agreement with experimental results in particular cases, there is no general agreement in the machining literature as to an appropriate model for the tool wear process.

In view of these facts, we have decided to use only the basic structural features of the tool wear model (1) in our approach, and to assume that additional details concerning the model are not available at present. More specifically, consider the following general model for the tool wear dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{2a}$$

$$F = \mathbf{a}^T\mathbf{u} + \mathbf{b}^T\mathbf{x} \tag{2b}$$

where $\mathbf{x} = [w_{f_1} \quad w_{f_2}]^T \in \Re^2$ is the tool wear state, $\mathbf{u} \in \Re^m$ is the vector of (reparameterized) system inputs such as feed, speed, and depth of cut, F is again the cutting force which is assumed to be measurable, $\mathbf{f} : \Re^2 \times \Re^m \rightarrow \Re^2$ quantifies the tool wear dynamics and is assumed to be unknown, and $\mathbf{a} \in \Re^m$, $\mathbf{b} = [b \quad b]^T \in \Re^2$ are vectors of unknown constants which quantify the output model. Note that the reparameterization of the inputs provides an affine (but unknown) relationship between the new inputs $\mathbf{u}$ and the output F. Observe that the model (2) incorporates only the basic qualitative features of the tool wear model (1) and does not assume that any quantitative model information is available.

The objective of this paper is to develop an algorithm which, given knowledge of the system inputs $\mathbf{u}$ and measurements of the system output F, can provide an accurate estimate of the system state $\mathbf{x}$. The proposed solution to this problem is to utilize a recurrent neural network to model the dynamical system (2) and a robust observer to estimate the tool wear state from this model using knowledge of $\mathbf{u}$ and F. The feasibility of this approach depends on the possibility of modeling (2) with a neural network; thus it will be helpful to establish a few facts concerning the approximation of dynamical systems using a recurrent neural network (RNN). Consider the RNN

$$\dot{\mathbf{x}}_n = -A\mathbf{x}_n + W\mathbf{z}(\mathbf{x}_n, \mathbf{u}) \tag{3}$$

where $\mathbf{x}_n \in \Re^n$ is the state of the network, $A = \text{diag}[a_i] \in \Re^{n\times n}$ with the a_i positive constants, $W \in \Re^{n\times p}$ is the matrix of adjustable weights for the network, and $\mathbf{z} : \Re^n \times \Re^m \rightarrow \Re^p$ is a function of the network states and inputs. The accuracy with which a RNN of the form (3) can approximate a general dynamical system such as (2a) depends crucially on the values used for the elements of W and on the functional form of $\mathbf{z}$. Several methods for selecting the form of $\mathbf{z}$ have been proposed in the neural network literature, and extensive theoretical, simulation and experimental studies have demonstrated the effectiveness of these methods. In what follows, we shall assume that $\mathbf{z}$ is constructed using either recurrent higher order neural network (RHONN) methods [e.g., 11] or Gaussian radial basis function network (GRBFN) techniques [e.g., 12]. The suitability of using either of these choices for $\mathbf{z}$ is indicated in the following lemma.

Lemma 1: Suppose that the RNN (3) represents either a RHONN or a GRBFN (depending on the choice of $\mathbf{z}$) and that the RNN (3) and the dynamical system (2a) are initially at the same state (i.e., $\mathbf{x}(0) = \mathbf{x}_n(0)$). Then for any $\epsilon > 0$ and any $T > 0$ there exists an integer p and a matrix W^* such that the network (3) with $W = W^*$ satisfies $\| \mathbf{x}(t) - \mathbf{x}_n(t) \| \leq \epsilon$ for all $t \in [0, T]$.

Proof: The proof is based on the Stone-Weierstrass theorem, and is given for RHONN in [11] and can be obtained for GRBFN through a straightforward generalization of the results in [11]. □

This result shows that, if a sufficient number of interconnections are permitted in the network, the dynamical system (2a) can be modeled with arbitrary accuracy using either RHONN or GRBFN models. Observe that this is an existence result, however, and that it does not prescribe a method for obtaining a weighting matrix W^* which ensures that the desired accuracy is obtained. This latter problem is considered in the following section, where the proposed tool wear estimation algorithm is developed.

Finally, we note that our approach to the design and analysis of both the RNN and observer components of the tool wear estimation algorithm is based on Lyapunov stability theory. The following lemma establishes a useful result concerning a certain class of Lyapunov functions and will be of direct relevance in this development.

Lemma 2: Consider the coupled dynamical system $\dot{\mathbf{x}}_1 = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, t)$, $\dot{\mathbf{x}}_2 = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, t)$. Let $V(\mathbf{x}_1, \mathbf{x}_2, t)$ be a Lyapunov function for the system with the properties

$$\lambda_1 \| \mathbf{x}_1 \|^2 + \lambda_2 \| \mathbf{x}_2 \|^2 \leq V \leq \lambda_3 \| \mathbf{x}_1 \|^2 + \lambda_4 \| \mathbf{x}_2 \|^2$$

$$\dot{V} \leq -\lambda_5 \| \mathbf{x}_1 \|^2 - \lambda_6 \| \mathbf{x}_2 \|^2 + \epsilon$$

where ϵ and the λ_i are positive scalar constants. Define $\delta = \max(\lambda_3/\lambda_5, \lambda_4/\lambda_6)$ and

$r_i = (\delta\epsilon/\lambda_i)^{1/2}$ for $i = 1, 2$. Then for any initial state $\mathbf{x}_1(0), \mathbf{x}_2(0)$ the system will evolve so that $\mathbf{x}_1(t), \mathbf{x}_2(t)$ are uniformly bounded and converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where $B_{r_i} = \{\mathbf{x}_i \ : \ \| \mathbf{x}_i \| \leq r_i\}$ (see [13] for a discussion of exponential convergence to a closed ball).

Proof: The proof is based on an extension of the global exponential convergence theorem of Corless [13] and is given in [14]. □

3. Tool Wear Estimation Algorithm

The proposed tool wear estimation algorithm consists of two components: a RNN and a robust observer. In what follows, each of these subsystems is described and analyzed in detail. The performance of the complete system is then illustrated through a case study in Section 4.

3.1 Neural Network Model for Tool Wear Dynamics

Consider the problem of approximating the dynamical system (with outputs) given in (2) using a RNN. First note that modeling the output equation (2b) requires only that the unknown constant vectors $\mathbf{a}, \mathbf{b}$ be determined. Since (2b) is linear in these unknown parameters, there are many techniques available for estimating them and thus for approximating (2b). For example, suppose that $\widehat{\mathbf{a}} \in \Re^m$ and $\widehat{\mathbf{b}} \in \Re^2$ are estimates for $\mathbf{a}$ and $\mathbf{b}$, and that these estimates are computed adaptively by comparing the evolution of the output $F(t)$ of the actual system (2) with the "predicted" output $\widehat{F}(t) = \widehat{\mathbf{a}}\mathbf{u} + \widehat{\mathbf{b}}\mathbf{x}$. One means of calculating these estimates based on the discrepancy between $F(t)$ and $\widehat{F}(t)$ is to utilize the following adaptive identification scheme:

$$\begin{aligned} \dot{\widehat{\mathbf{a}}} &= \gamma(F - \widehat{F})\mathbf{u} \\ \dot{\widehat{\mathbf{b}}} &= \gamma(F - \widehat{F})\mathbf{x} \end{aligned} \tag{4}$$

where γ is a positive scalar constant. The suitability of this adaptive approach to estimating $\mathbf{a}, \mathbf{b}$ is indicated in the following lemma.

Lemma 3: Suppose that the input $\mathbf{u}(t)$ to the dynamical system (2) is chosen so that

$$\int_t^{t+T} \mathbf{y}(\tau)\mathbf{y}^T(\tau)d\tau \geq \alpha I_{m+2}$$

for all $t \geq 0$, where $\mathbf{y} = [\mathbf{u}^T \ \ \mathbf{x}^T]^T \in \Re^{m+2}$, I_{m+2} is the $(m+2) \times (m+2)$ identity matrix, and T, α are positive constants. Then the estimation scheme (4) ensures that $\widehat{\mathbf{a}} \to \mathbf{a}$ and $\widehat{\mathbf{b}} \to \mathbf{b}$ as $t \to \infty$ for any initial $\widehat{\mathbf{a}}(0)$ and $\widehat{\mathbf{b}}(0)$.

Proof: Let $\mathbf{c} = [\mathbf{a}^T \;\; \mathbf{b}^T]^T$ and $\widehat{\mathbf{c}} = [\widehat{\mathbf{a}}^T \;\; \widehat{\mathbf{b}}^T]^T$, so that $F = \mathbf{c}^T\mathbf{y}$ and $\widehat{F} = \widehat{\mathbf{c}}^T\mathbf{y}$. Consider the Lyapunov function candidate $V_1 = \phi^T\phi/2$, where $\phi = \mathbf{c} - \widehat{\mathbf{c}}$. Differentiating V_1 along the trajectories of (4) yields, after simplification

$$\begin{aligned} \dot{V}_1 &= -\gamma\phi^T(F - \widehat{F})\mathbf{y} \\ &= -\gamma\phi^T\mathbf{y}\mathbf{y}^T\phi \end{aligned}$$

Standard arguments from differential equation stability theory then show that, if the condition on $\mathbf{u}$ given in the statement of the lemma is satisfied, then $\phi \to \mathbf{0}$ as $t \to \infty$ and thus the desired estimates for $\mathbf{a}, \mathbf{b}$ are obtained [15]. □

Using the preceding result, it is straightforward to obtain accurate estimates for $\mathbf{a}$ and $\mathbf{b}$; for example, it is relatively easy to ensure that the condition on $\mathbf{y}$ stated in Lemma 3 is satisfied during a "training" period for the algorithm. Thus it what follows it is assumed that $\mathbf{a}$ and $\mathbf{b}$ have been identified in this manner, so that the problem of modeling the dynamical system (2) using a RNN reduces to the problem of approximating (2a) using such a network. This approximation process involves selecting the RNN weights W and functions $\mathbf{z}$ in such a way that the response of the RNN closely models the response of (2a) if each system is subjected to the same inputs. Lemma 1 indicates that an appropriate choice for $\mathbf{z}$ can be obtained using either RHONN methods or GRBFN techniques. Determining suitable values for the elements of W is ordinarily accomplished by presenting a set of input trajectories to both the dynamical system to be modeled and the RNN, and adjusting the weighting matrix W in such a way that the discrepancy between the state trajectories for the two systems is reduced; this process is often termed "training" the RNN. In what follows, we propose a robust training procedure for the RNN given in (3).

It can be seen from Lemma 1 that, if $\mathbf{z}$ is chosen so that (3) is in RHONN or GRBFN (with sufficient interconnections p) then there exists a choice for W, say W^*, for which (3) provides a close approximation of (2a). In other words, given this choice for W, the dynamical system (2a) can be written as

$$\dot{\mathbf{x}} = -A\mathbf{x} + W^*\mathbf{z}(\mathbf{x}, \mathbf{u}) + \eta(\mathbf{x}, \mathbf{u}) \tag{5}$$

where $\eta : \Re^2 \times \Re^m \to \Re^2$ quantifies the (small) modeling error. The difficulty, of course, is determining W^*. A robust approach for specifying W, and thereby training the RNN, is proposed in the following theorem.

Theorem 1: Consider the following strategy for determining W:

$$\dot{\mathbf{x}}_n = -A\mathbf{x}_n + W\mathbf{z}(\mathbf{x}, \mathbf{u}) \tag{6a}$$

$$\dot{W} = -\alpha W\,1\!\!+\!70\beta\mathbf{e}\mathbf{z}^T \tag{6b}$$

where $\mathbf{e} = \mathbf{x} - \mathbf{x}_n$ and α, β are positive scalar constants. This strategy ensures that $\mathbf{e}$ and $\widetilde{W} = W^* - W$ are uniformly bounded for all model mismatches η, and that the ultimate bound on $\mathbf{e}$ is small if η is small.

Proof: The identification error dynamics can be obtained by differencing (5) and (6a):

$$\dot{\mathbf{e}} = -A\mathbf{e} + \widetilde{W}\mathbf{z} + \eta \tag{7}$$

Consider the Lyapunov function candidate

$$V_2 = \frac{1}{2}\mathbf{e}^T\mathbf{e} + \frac{1}{2\beta}\mathrm{tr}[\widetilde{W}\widetilde{W}^T] \tag{8}$$

Differentiating (8) along (7) and simplifying yields

$$\begin{aligned} \dot{V}_2 &= -\mathbf{e}^T A\mathbf{e} + \mathbf{e}^T\eta + \frac{1}{\beta}\mathrm{tr}[\widetilde{W}(\dot{\widetilde{W}} + \beta\mathbf{e}\mathbf{z}^T)^T] \\ &\leq -a_{min} \parallel \mathbf{e} \parallel^2 + \eta_{max} \parallel \mathbf{e} \parallel - \frac{\alpha}{\beta} \parallel \widetilde{W} \parallel_F^2 + \frac{\alpha}{\beta} \parallel W^* \parallel_F \parallel \widetilde{W} \parallel_F \end{aligned}$$

where $a_{min} = \min(a_i)$, η_{max} is an upper bound on $\parallel \eta \parallel$, and $\parallel \cdot \parallel_F$ is the Frobenius norm. Routine manipulation allows the following upper bound on $\dot{V}_2$ to be derived:

$$\dot{V}_2 \leq -\frac{a_{min}}{2} \parallel \mathbf{e} \parallel^2 - \frac{\alpha}{2\beta} \parallel \widetilde{W} \parallel_F^2 + \frac{\eta_{max}^2}{2a_{min}} + \frac{\alpha \parallel W^* \parallel_F^2}{2\beta}$$

Application of Lemma 2 then permits the conclusion that $\mathbf{e}, \widetilde{W}$ are uniformly bounded and are guaranteed to converge exponentially to a compact neighborhood of the origin in $\mathbf{e}, \widetilde{W}$ space. A bound on the ultimate size of $\parallel \mathbf{e} \parallel$ can also be obtained using Lemma 2; if β is chosen reasonably large then the size of this ultimate bound is proportional to the size of the model mismatch η_{max}, so that $\mathbf{e}$ is small if η is small. □

The results summarized in Lemma 1, Lemma 3, and Theorem 1 define a strategy for modeling the uncertain tool wear dynamical system (2) using a RNN together with an (adaptively obtained) output map. If the conditions stipulated in Lemma 1 and Theorem 1 are satisfied, the response of the state $\mathbf{x}_n$ of the RNN to a given input trajectory should provide an accurate approximation of the response of the actual system to this input trajectory. Thus this RNN could, in principle, be used to give an estimate for tool wear during machining. However, it is highly desirable to introduce a feedback mechanism into this estimation algorithm, so that the wear estimate is influenced by the evolution of the actual system. One means of accomplishing this goal is described in the next section.

3.2 Robust Observer for Tool Wear

Let $W_c \in \Re^{n \times p}$ denote the matrix of constants obtained by training the RNN (3) using the robust identification scheme summarized in Theorem 1. If the results given in Lemma 1, Lemma 3, and Theorem 1 are used to construct the RNN model of the tool wear dynamics (2), then this dynamical system can be rewritten as

$$\dot{\mathbf{x}} = -A\mathbf{x} + W_c\mathbf{z}(\mathbf{x}, \mathbf{u}) + \eta(\mathbf{x}, \mathbf{u}) \tag{9a}$$

$$F^* = \mathbf{b}^T\mathbf{x} \tag{9b}$$

where $F^* = F - \mathbf{a}^T\mathbf{u}$. In (9), η is unknown but small and $A, W_c, \mathbf{z}, \mathbf{b}$, and F^* are known (observe, for example, that F^* is known because F can be measured, $\mathbf{u}$ is known and $\mathbf{a}$ has been identified).

Our objective is now to develop an observer which can asymptotically reconstruct the tool wear state $\mathbf{x}$ from measurements of F^* despite the system uncertainty quantified by η. Note first that the original system (2) is observable [9], so that this objective is attainable. Consider the observer

$$\dot{\mathbf{x}}_o = -A\mathbf{x}_o + W_c\mathbf{z}(\mathbf{x}_o, \mathbf{u}) + L(F^* - \mathbf{b}^T\mathbf{x}_o) \tag{10}$$

where $\mathbf{x}_o \in \Re^2$ is the estimate of the wear state $\mathbf{x}$ and $L \in \Re^{2\times 1}$ is the observer gain matrix to be specified later. Applying (10) to (9) yields

$$\dot{\mathbf{e}} = -(A + L\mathbf{b}^T)\mathbf{e} + W_c[\mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o)] + \eta \tag{11}$$

where $\mathbf{e} = \mathbf{x} - \mathbf{x}_o$ is the state estimation error and the dependence of $\mathbf{z}$ on $\mathbf{u}$ has been supressed for clarity. The performance characteristics of the observer (10) can be determined by studying the stability properties of (11). The following fact concerning the pair of matrices $(A, \mathbf{b})$ will be helpful in this regard.

Lemma 4: If the RNN is designed so that the diagonal elements of A are not equal (this is always possible) then the eigenvalues of $(A + L\mathbf{b}^T)$ can be arbitrarily assigned through the specification of L.

Proof: It is trivial to verify that the matrix $[\mathbf{b}^T \;\; A^T\mathbf{b}]$ has full rank, and the assignability of the eigenvalues of $(A + L\mathbf{b}^T)$ follows immediately from this [e.g., 16]. □

The following theorem characterizes the performance capabilities of the observer (10).

Theorem 2: The observer (10) ensures that the state estimation error $\mathbf{e}$ is uniformly bounded, and has an ultimate bound which can be reduced arbitrarily, provided that L is properly specified.

Proof: Suppose that L is chosen so that $\lambda = \lambda_{min}(A + L\mathbf{b}^T) = \lambda_{max}(A + L\mathbf{b}^T)/2$ is a large positive constant. Let $\Lambda = \mathrm{diag}(\lambda \;\; 2\lambda)$ and M^{-1} denote the associated (normalized) modal matrix, so that $\Lambda = M(A + L\mathbf{b}^T)M^{-1}$. Then defining $\mathbf{q} = M\mathbf{e}$ permits (11) to be rewritten

$$\dot{\mathbf{q}} = -\Lambda\mathbf{q} + MW_c[\mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o)] + M\eta \qquad (12)$$

Consider the Lyapunov function candidate $V_3 = \mathbf{q}^T\mathbf{q}/2$. Differentiating V_3 along the trajectories of (12) yields, after simplification

$$\begin{aligned}\dot{V}_3 &\leq -\lambda \parallel \mathbf{q} \parallel^2 + \parallel M \parallel_F \parallel \eta \parallel \parallel \mathbf{q} \parallel + \parallel M \parallel_F \parallel W_c \parallel_F \parallel \mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o) \parallel \parallel \mathbf{q} \parallel \\ &\leq -(\lambda - K \parallel M \parallel_F \parallel W_c \parallel_F) \parallel \mathbf{q} \parallel^2 + \parallel M \parallel_F \parallel \eta \parallel \parallel \mathbf{q} \parallel\end{aligned}$$

where $K > 0$ satisfies $K \parallel M(\mathbf{x} - \mathbf{x}_o) \parallel \geq \parallel \mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o) \parallel \;\; \forall \mathbf{x}, \mathbf{x}_o$ (the existence of such a K is a consequence of the fact that, for RHONN and GRBFN designs, the partial derivatives of $\mathbf{z}$ are all bounded). Assume that L is chosen so that $\lambda_0 = \lambda - K \parallel M \parallel_F \parallel W_c \parallel_F$ is positive (this is always possible). Then routine manipulation permits $\dot{V}_3$ to be upper bounded as

$$\dot{V}_3 \leq -(\lambda_0 \parallel \mathbf{q} \parallel - \eta_{max} \parallel M \parallel_F) \parallel \mathbf{q} \parallel$$

From Lemma 2 it can now be concluded that $\parallel \mathbf{q} \parallel$ is uniformly bounded and that the ultimate bound on this error is $\parallel \mathbf{q} \parallel \leq \eta_{max} \parallel M \parallel_F / \lambda_0$; note that the ultimate bound on $\parallel \mathbf{q} \parallel$ (and therefore $\parallel \mathbf{e} \parallel$) can be decreased as desired by increasing λ_0 through the specification of L. □

The observer (10) can be modified to provide improved performance for those cases in which the RNN is designed so that the model mismatch η possesses some structure. For example, suppose that the RNN and the observer gain matrix L are selected so that $M^TM\eta = [\eta_1 \;\; \eta_2]^T$ with $\eta_1 \approx \eta_2$ [17]. In this situation, if

$$\eta^* = \frac{1}{2}\begin{bmatrix}\eta_1 + \eta_2 \\ \eta_1 + \eta_2\end{bmatrix}$$

$$\eta^{**} = \frac{1}{2}\begin{bmatrix}\eta_1 - \eta_2 \\ \eta_2 - \eta_1\end{bmatrix}$$

then $M^TM\eta = \eta^* + \eta^{**}$ and η^{**} is very small. A modified observer can then be developed which exploits this additional information on the stucture of the model mismatch; this modification is proposed in the following theorem.

Theorem 3: Consider the observer

$$\dot{\mathbf{x}}_o = -A\mathbf{x}_o + W_c\mathbf{z}(\mathbf{x}_o, \mathbf{u}) + L(F^* - \mathbf{b}^T\mathbf{x}_o) + \mathbf{s}(\mathbf{x}_o, F^*) \qquad (13a)$$

where all terms are defined as in (10) and

$$\mathbf{s} = \begin{cases} (M^T M)^{-1}\mathbf{b}\frac{\mathbf{b}^T\mathbf{e}}{|\mathbf{b}^T\mathbf{e}|}\rho & \text{if } \mathbf{b}^T\mathbf{e} \neq 0 \\ \mathbf{0} & \text{if } \mathbf{b}^T\mathbf{e} = 0 \end{cases} \tag{13b}$$

with $\rho > 0$. Note that, although $\mathbf{e}$ appears in the definition for $\mathbf{s}$ in (13b), $\mathbf{s}$ can be computed using only knowledge of $\mathbf{x}_o$ and F^* so that (13) is implementable. The observer (13) ensures that the state estimation error $\mathbf{e}$ is uniformly bounded provided that L and ρ are properly specified. Moreover, if η^{**} is negligible then $\mathbf{e}$ converges to zero exponentially.
Proof: The observer (13) leads to the following state estimation error dynamics (in the error coordinate $\mathbf{q}$)

$$\dot{\mathbf{q}} = -\Lambda\mathbf{q} + MW_c[\mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o)] + M[(M^T M)^{-1}\eta^* - \mathbf{s}(\mathbf{x}_o, F^*)] \tag{14}$$

where it is assumed that η^{**} is negligible. Differentiating the Lyapunov function candidate V_3 defined in Theorem 2 along the trajectories of (14) and simplifying yields the following upper bound on $\dot{V}_3$:

$$\begin{aligned} \dot{V}_3 &\leq -(\lambda - K \parallel M \parallel_F \parallel W_c \parallel_F) \parallel \mathbf{q} \parallel^2 + \mathbf{e}^T\eta^* - |\mathbf{b}^T\mathbf{e}|\rho \\ &\leq -(\lambda - K \parallel M \parallel_F \parallel W_c \parallel_F) \parallel \mathbf{q} \parallel^2 + \mathbf{e}^T\mathbf{b}\eta^{***} - |\mathbf{b}^T\mathbf{e}|\rho \end{aligned}$$

where $\eta^{***} = (\eta_1 + \eta_2)/2b$ and all the other terms are as defined in the proof of Theorem 2. Thus if ρ is chosen so that $\rho > \eta^{***} \;\; \forall\, \mathbf{x}, \mathbf{u}$ then

$$\begin{aligned} \dot{V}_3 &\leq -(\lambda - K \parallel M \parallel_F \parallel W_c \parallel_F) \parallel \mathbf{q} \parallel^2 - |\mathbf{b}^T\mathbf{e}|(\rho - \eta^{***}) \\ &\leq -(\lambda - K \parallel M \parallel_F \parallel W_c \parallel_F) \parallel \mathbf{q} \parallel^2 \end{aligned}$$

and the exponential convergence to the origin of $\mathbf{q}$ (and therefore $\mathbf{e}$) follows from Lemma 2 (with $\epsilon = 0$). □

4. Case Study: Turning

The proposed tool wear estimation algorithm was evaluated through a case study involving a turning application. In this case study, AISI medium carbon steel cylinders having 223 BHN hardness were turned using a CNC lathe with Valenite VC55 TNMA432 uncoated cemented carbide tools. Cutting force measurements were obtained for all trials, and the following cutting conditions were used throughout the case study: cutting speed = 200m/min, feed = 0.15mm/rev, and depth of cut = 1.27mm.

The tool wear estimation algorithm utilized in this case study consisted of an RHONN of the form given in (3) together with the state observer (10). It is noted that the observer

(13) was also implemented for several trials and provided results which were indistinguishable from those obtained using (10), so that only the results using (10) are reported here. The RHONN structure was determined using the design procedure given in [11]. The network was trained using data generated through computer simulation using the tool wear model proposed in [10]. In each trial only two cutting experiments were simulated, so that the training data sets were quite modest. The parameter values used in the observer (10) were obtained after a few iterations using computer simulation data, and no attempt was made to "tune" the parameters to achieve optimal performance. Instead, two sets of observer parameters were used: a "low gain" observer which resulted in a moderate magnitude for λ, and a "high gain" observer which placed the magnitude of λ at a large value. All details concerning the implementation of the proposed estimation algorithm, including the structure of the RHONN and the parameter values for the network and the observer, are given in [18].

To evaluate the performance of the "trained" estimation algorithm, the algorithm was tested using the experimental data obtained by Park and Ulsoy in [9]. More specifically, the force measurement trajectories reported in [9] were used as inputs to the wear state estimation algorithm to generate estimated tool wear trajectories. The estimates for the flank wear components due to abrasion (w_{f_1}) and diffusion (w_{f_2}) for typical trials with the low gain observer and high gain observer are shown in Figures 1a and 1b, respectively. Note that, although the estimates of the individual components of tool wear obtained using the high gain observer are the more reasonable (the low gain observer estimates a *reduction* in wear due to diffusion in this trial), the estimate of total tool wear $w_f = w_{f_1} + w_{f_2}$ is comparable in each case. The evolution of the total tool wear w_f obtained using the high gain observer is shown in Figure 2, and it can be seen that this estimate agrees quite closely with the experimental measurements made by Park and Ulsoy [9].

5. Conclusions

This paper presents a robust strategy for estimating tool wear in metal cutting operations. The proposed estimation algorithm consists of a recurrent neural network to model the tool wear dynamics, and a state observer to estimate the tool wear from this model using measurements of cutting force. A stability analysis is performed to demonstrate that the algorithm provides uniformly bounded tool wear estimation error and that the ultimate bound on this error can be reduced arbitrarily. The proposed approach is applied to the problem of estimating tool wear in turning and is shown to provide wear estimates which are in close agreement with experimental results. Future work will focus on applying the proposed approach of real-time state estimation to other problems of importance in manufacturing.

6. Acknowledgments

The research described in this paper was supported in part through contracts with the Army Research Office and the Jet Propulsion Laboratory.

7. References

1. Matsushima, K. and T. Sata, "Development of Intelligent Machine Tool" *Journal of the Faculty of Engineering*, University of Tokyo, Vol. 35, No. 3, 1980

2. Rangwala, S., S. Liang, and D. Dornfeld, "Pattern Recognition of Acoustic Emission Signals During Punch Stretching" *Mechanical Systems and Signal Processing*, Vol. 1, No. 4, 1987, pp. 321-332

3. Rangwala, S., "Machining Process Characterization and Intelligent Tool Condition Monitoring Using Acoustic Emission Signal Analysis" Ph.D. thesis, University of California, Berkeley, 1988

4. Burke, L., "An Unsupervised Neural Network Approach to Tool Wear Identification" *IIE Transactions*, Vol. 25, No. 1, 1993, pp. 16-25

5. Yao, Y. and X. Fang, "Assessment of Chip Forming Patterns with Tool Wear Progression in Machining via Neural Networks" *International Journal of Machine Tools Manufacture*, Vol. 33, No. 1, 1993, pp. 89-102

6. Koren, Y., "Flank Wear Model of Cutting Tools Using Control Theory", *ASME Journal of Engineering for Industry*, Vol. 100, No. 1, 1978, pp. 103-109

7. Danai, K. and A. Ulsoy, "A Dynamic State Model for On-Line Tool Wear Estimation in Turning", *ASME Journal of Engineering for Industry*, Vol. 109, No. 4, 1987, pp. 396-399

8. Koren, Y., T. Ko, K. Danai, and A. Ulsoy, "Flank Wear Estimation Under Varying Cutting Conditions", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 113, No. 2, 1991, pp. 300-307

9. Park, J. and A. Ulsoy, "On-Line Tool Wear Estimation Using Force Measurement and a Nonlinear Observer", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 114, No. 4, 1992, pp. 666-672

10. Park, J. and A. Ulsoy, "On-Line Flank Wear Estimation Using an Adaptive Observer and Computer Vision: Theory and Experiment", *ASME Journal of Engineering for Industry*, Vol. 115, No. 1, 1993, pp. 30-43

11. Kosmatopoulos, E., P. Ioannou, and M. Christodoulou, "Identification of Nonlinear Systems Using New Dynamic Neural Network Structures" *Proc. IEEE Conference on Decision and Control*, Tucson, AZ, December 1992

12. Sanner, R. and J.-J. Slotine, "Gaussian Networks for Direct Adaptive Control" *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, 1992, pp. 837-863

13. Corless, M., "Guaranteed Rates of Exponential Convergence for Uncertain Systems", *Journal of Optimization Theory and Applications*, Vol. 64, No. 3, 1990, pp. 481-494

14. Colbaugh, R., H. Seraji, and K. Glass, "Adaptive Compliant Motion Control for Dexterous Manipulators", *International Journal of Robotics Research*, Vol. 14, No. 3, 1995 (in press)

15. Vidyasagar, M., *Nonlinear Systems Analysis*, Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1993

16. Sontag, E., *Mathematical Control Theory*, Springer-Verlag, New York, 1990

17. Walcott, B., M. Corless, and S. Zak, "Comparative Study of Non-Linear State-Observation Techniques", *International Journal of Control*, Vol. 45, No. 6, 1987, pp. 2109-2132

18. Colbaugh, R. and K. Glass, "A Robust Approach to Real-Time Tool Wear Estimation", Robotics Laboratory Report, New Mexico State University, Las Cruces, NM, December 1993

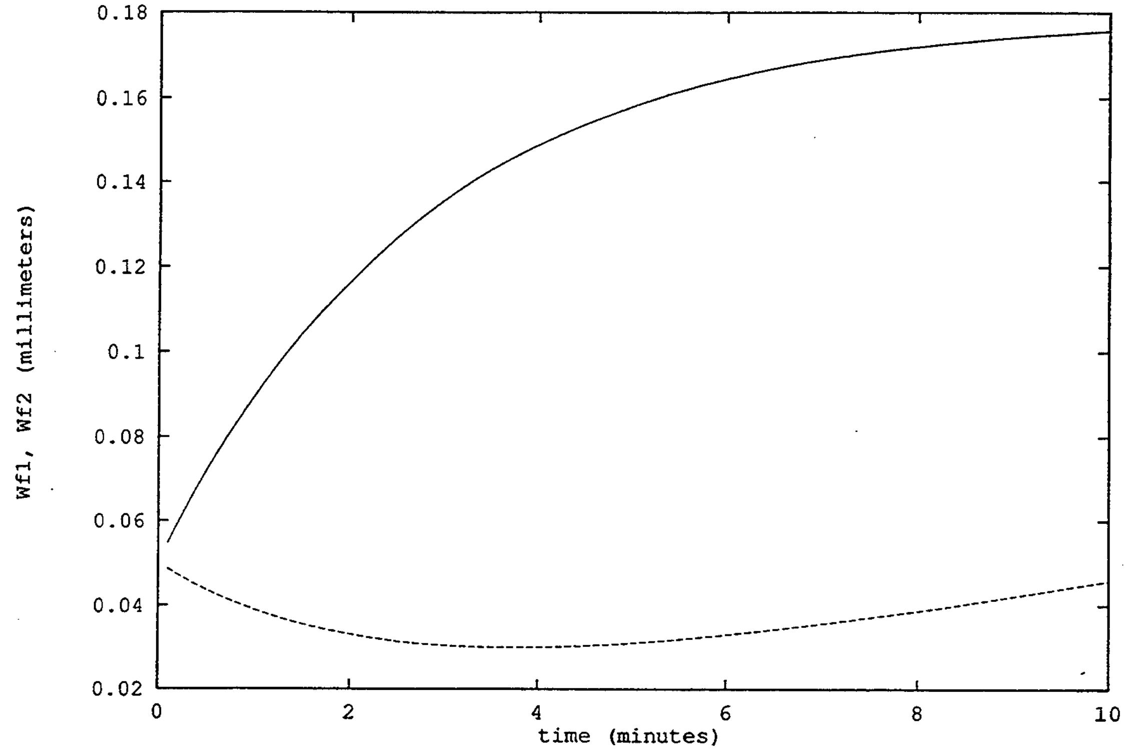


Figure 1a: Estimated flank wear components (low gain observer)

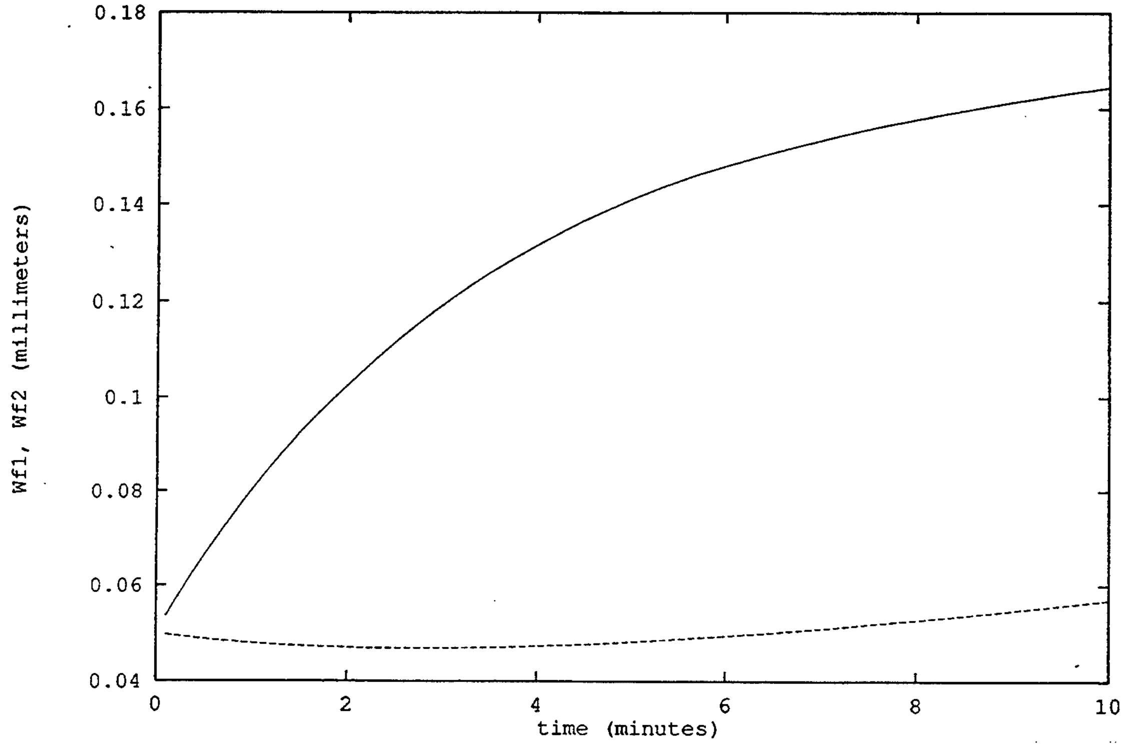


Figure 1b: Estimated flank wear components (high gain observer)

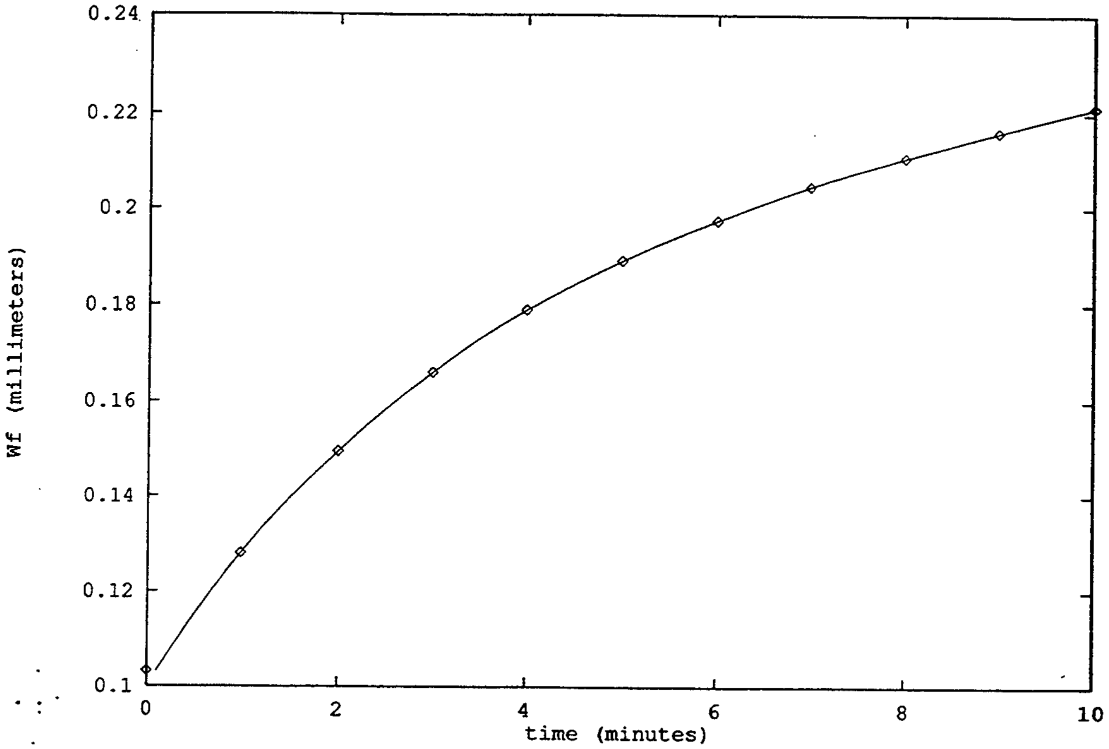


Figure 2: Estimated and measured flank wear (high gain observer)

to appear: Proc. IEEE International Conference on Robotics and Automation, Minneapolis, MN, April 1996

REAL-TIME TOOL WEAR ESTIMATION USING CUTTING FORCE MEASUREMENTS

K. Glass **R. Colbaugh**

Department of Mechanical Engineering
New Mexico State University, Las Cruces, NM 88003 USA

Abstract

This paper presents a robust strategy for estimating tool wear in metal cutting operations. The proposed estimation algorithm consists of two components: a recurrent neural network to model the tool wear dynamics, and a robust observer to estimate the tool wear from this model using measurements of cutting force. It is shown that the algorithm ensures that the tool wear estimation error is uniformly bounded in the presence of bounded unmodeled effects, and that the ultimate bound on this error can be made as small as desired. The proposed approach is applied to the problem of estimating tool wear in turning and is shown to provide wear estimates which are in close agreement with published experimental results.

1. Introduction

There is considerable interest in increasing the level of automation in manufacturing processes. Included among the anticipated benefits of this automation are enhanced productivity, improved product quality, and reduced manufacturing costs. However, progress toward the development of automated manufacturing systems has been hindered by the lack of dependable machine tool diagnostic schemes. In particular, the problem of reliably measuring the tool wear has been a significant obstacle to the realization of effective automated machining systems. The approaches proposed thusfar for measuring tool wear can be divided into two broad classes: direct methods and indirect methods. In the direct approach, tool wear is measured directly using, for example, microscopes or computer vision technology. While this method can provide very accurate wear information, it is inherently an off-line approach and is therefore limited in its applicability. Alternatively, the indirect approach to tool wear measurement is to estimate the wear state from measurable cutting variables, such as cutting force or acoustic emission. Observe that the indirect method can be used for real-time tool wear determination and consequently has considerable potential for use in automated machining systems. However, the accuracy of existing indirect techniques is often unsatisfactory.

Recognizing the importance of reliably estimating the machine tool wear state in real-time, several researchers have studied this problem in recent years [e.g., 1-10]. Early work focused on the application of statistical methods to associate patterns in measurable signals with wear states [e.g., 1,2]. More recently, it has been proposed that this pattern recognition could be accomplished effectively using neural networks [3-5]. A different approach to tool wear estimation is to acknowledge that tool wear evolves as a dynamical system during the cutting process, and to apply system theoretic ideas to the estimation of wear state [e.g., 6-10]. The principle difficulty associated with this strategy is the nonlinear nature of the process model and the lack of good estimates for the model parameter values, so that traditional state observation techniques are difficult to apply. Nevertheless, this method has considerable potential because the formulation is compatible with the dynamic character of the tool wear process.

This paper presents a new strategy for the real-time estimation of tool wear in metal cutting operations. The proposed estimation algorithm consists of two components: a recurrent neural network to model the tool wear dynamics, and a robust observer to estimate the tool wear from this model using measurements of cutting force. Thus the approach is to train the neural network so that it provides a good model for the nonlinear and uncertain tool wear dynamical system, and then to design the observer to ensure good performance of the overall system in the presence of (inevitable) modeling error. A stability analysis is conducted to show that tool wear estimation error is uniformly bounded with an ultimate bound that can be reduced arbitrarily. The efficacy of the proposed approach is demonstrated through a case study involving the estimation of tool wear in a turning application; it is shown in this study that the algorithm provides wear estimates which are in close agreement with published experimental results.

2. Preliminaries

This paper considers the problem of estimating in real-time the tool wear state during machining operations. Numerous researchers have studied the cutting process and proposed models for tool wear. For example, Park and Ulsoy [9] have proposed that flank wear dominates the tool wear process in many important applications, and have proposed the following model for this component of tool wear:

$$\dot{w}_{f_1} = -k_1 v w_{f1} + k_2 \frac{Fv}{fd}$$
$$\dot{w}_{f_2} = k_3 \sqrt{v} \exp(\frac{-k_4}{T}) \quad (1)$$
$$F = k_5 f^n d + k_6 v d + k_7 d + k_8 d(w_{f_1} + w_{f_2})$$

where w_{f_1} and w_{f_2} are the flank wear components due to abrasion and diffusion, respectively, $w_f = w_{f_1} + w_{f_2}$ is the total flank wear, f, v, d are the feed, cutting speed, and depth of cut, respectively, and are the inputs for the system, and F is the cutting force and is the output (measurable variable) for the system. In (1), n and the k_i are constants which depend on the effective rake angle, workpiece material, and other task-related parameters, and T is the tool-work temperature and depends on the inputs f, v and the state w_{f_1}, w_{f_2}. Observe that the tool wear model (1) is a complicated and highly uncertain dynamical system. For example, the model for tool-work temperature is purely an empirical one, and only crude estimates are available for the values of most of the parameters which appear in (1). Indeed, although the model (1) has shown reasonable agreement with experimental results in particular cases, there is no general agreement in the machining literature as to an appropriate model for the tool wear process.

In view of these facts, we have decided to use only the basic structural features of the tool wear model (1) in our approach, and to assume that additional details concerning the model are not available at present. More specifically, consider the following general model for the tool wear dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2a)$$
$$F = \mathbf{a}^T \mathbf{u} + \mathbf{b}^T \mathbf{x} \quad (2b)$$

where $\mathbf{x} = [w_{f_1} \quad w_{f_2}]^T \in \Re^2$ is the tool wear state, $\mathbf{u} \in \Re^m$ is the vector of (reparameterized) system inputs such as feed, speed, and depth of cut, F is again the cutting force which is assumed to be measurable, $\mathbf{f} : \Re^2 \times \Re^m \rightarrow \Re^2$ quantifies the tool wear dynamics and is assumed to be unknown, and $\mathbf{a} \in \Re^m$, $\mathbf{b} = [b \quad b]^T \in \Re^2$ are vectors of unknown constants which quantify the output model. Note that the reparameterization of the inputs provides an affine (but unknown) relationship between the new inputs $\mathbf{u}$ and the output F. Observe that the model (2) incorporates only the basic qualitative features of the tool wear model (1) and does not assume that any quantitative model information is available.

The objective of this paper is to develop an algorithm which, given knowledge of the system inputs $\mathbf{u}$ and measurements of the system output F, can provide an accurate estimate of the system state $\mathbf{x}$. The proposed solution to this problem is to utilize a recurrent neural network to model the dynamical system (2) and a robust observer to estimate the tool wear state from this model using knowledge of $\mathbf{u}$ and F. The feasibility of this approach depends on the possibility of modeling (2) with a neural network; thus it will be helpful to establish a few facts concerning the approximation of dynamical systems using a recurrent neural network (RNN). Consider the RNN

$$\dot{\mathbf{x}}_n = -A\mathbf{x}_n + W\mathbf{z}(\mathbf{x}_n, \mathbf{u}) \quad (3)$$

where $\mathbf{x}_n \in \Re^n$ is the state of the network, $A = \mathrm{diag}[a_i] \in \Re^{n \times n}$ with the a_i positive constants, $W \in \Re^{n \times p}$ is the matrix of adjustable weights for the network, and $\mathbf{z} : \Re^n \times \Re^m \rightarrow \Re^p$ is a function of the network states and inputs. The accuracy with which a RNN of the form (3) can approximate a general dynamical system such as (2a) depends crucially on the values used for the elements of W and on the functional form of $\mathbf{z}$. Several methods for selecting the form of $\mathbf{z}$ have been proposed in the neural network literature, and extensive theoretical, simulation, and experimental studies have demonstrated the effectiveness of these methods. In what follows, we shall assume that $\mathbf{z}$ is constructed using either recurrent higher order neural network (RHONN) methods [e.g., 11] or Gaussian radial basis function network (GRBFN) techniques [e.g., 12]. The suitability of using either of these choices for $\mathbf{z}$ is indicated in the following lemma.

Lemma 1: Suppose that the RNN (3) represents either a RHONN or a GRBFN (depending on the choice of $\mathbf{z}$) and that the RNN (3) and the dynamical system (2a) are initially at the same state (i.e., $\mathbf{x}(0) = \mathbf{x}_n(0)$). Then for any $\epsilon > 0$ and any $T > 0$ there exists an integer p and a matrix W^* such that the network (3) with $W = W^*$ satisfies $\| \mathbf{x}(t) - \mathbf{x}_n(t) \| \leq \epsilon$ for all $t \in [0, T]$.
Proof: The proof is based on the Stone-Weierstrass theorem, and is given for RHONN in [11] and can be obtained for GRBFN through a straightforward generalization of the results in [11]. □

This result shows that, if a sufficient number of interconnections are permitted in the network, the dynamical system (2a) can be modeled with arbitrary accuracy using either RHONN or GRBFN models. Observe that this is an existence result, however, and that it does not prescribe a method for obtaining a weighting matrix W^* which ensures that the desired accuracy is obtained. This latter problem is considered in the following section, where the proposed tool wear estimation algorithm is developed.

Finally, we note that our approach to the design and analysis of both the RNN and observer components of the tool wear estimation algorithm is based on Lyapunov stability theory. The following lemma establishes a useful result concerning a certain class of Lyapunov functions and will be of direct relevance in this development.

Lemma 2: Consider the coupled dynamical system $\dot{\mathbf{x}}_1 = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, t)$, $\dot{\mathbf{x}}_2 = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, t)$. Let $V(\mathbf{x}_1, \mathbf{x}_2, t)$ be a Lyapunov function for the system with the properties

$$\lambda_1 \parallel \mathbf{x}_1 \parallel^2 + \lambda_2 \parallel \mathbf{x}_2 \parallel^2 \leq V \leq \lambda_3 \parallel \mathbf{x}_1 \parallel^2 + \lambda_4 \parallel \mathbf{x}_2 \parallel^2$$
$$\dot{V} \leq -\lambda_5 \parallel \mathbf{x}_1 \parallel^2 - \lambda_6 \parallel \mathbf{x}_2 \parallel^2 + \epsilon$$

where ϵ and the λ_i are positive scalar constants. Define $\delta = \max(\lambda_3/\lambda_5, \lambda_4/\lambda_6)$ and $r_i = (\delta\epsilon/\lambda_i)^{1/2}$ for $i = 1, 2$. Then for any initial state $\mathbf{x}_1(0), \mathbf{x}_2(0)$ the system will evolve so that $\mathbf{x}_1(t), \mathbf{x}_2(t)$ are uniformly bounded and converge exponentially to the closed balls B_{r_1}, B_{r_2}, respectively, where $B_{r_i} = \{\mathbf{x}_i : \parallel \mathbf{x}_i \parallel \leq r_i\}$ (see [13] for a discussion of exponential convergence to a closed ball).

Proof: The proof is based on an extension of the global exponential convergence theorem of Corless [13] and is given in [14]. □

3. Tool Wear Estimation Algorithm

The proposed tool wear estimation algorithm consists of two components: a RNN and a robust observer. In what follows, each of these subsystems is described and analyzed in detail. The performance of the complete system is then illustrated through a case study in Section 4.

3.1 Neural Network Model

Consider the problem of approximating the dynamical system (with outputs) given in (2) using a RNN. First note that modeling the output equation (2b) requires only that the unknown constant vectors $\mathbf{a}, \mathbf{b}$ be determined. Since (2b) is linear in these unknown parameters, there are many techniques available for estimating them and thus for approximating (2b). For example, suppose that $\widehat{\mathbf{a}} \in \Re^m$ and $\widehat{\mathbf{b}} \in \Re^2$ are estimates for $\mathbf{a}$ and $\mathbf{b}$, and that these estimates are computed adaptively by comparing the evolution of the output $F(t)$ of the actual system (2) with the "predicted" output $\widehat{F}(t) = \widehat{\mathbf{a}}\mathbf{u} + \widehat{\mathbf{b}}\mathbf{x}$. One means of calculating these estimates based on the discrepancy between $F(t)$ and $\widehat{F}(t)$ is to utilize the following adaptive identification scheme:

$$\dot{\widehat{\mathbf{a}}} = \gamma(F - \widehat{F})\mathbf{u}$$
$$\dot{\widehat{\mathbf{b}}} = \gamma(F - \widehat{F})\mathbf{x} \tag{4}$$

where γ is a positive scalar constant. The suitability of this adaptive approach to estimating $\mathbf{a}, \mathbf{b}$ is indicated in the following lemma.

Lemma 3: Suppose that the input $\mathbf{u}(t)$ to the dynamical system (2) is chosen so that

$$\int_t^{t+T} \mathbf{y}(\tau)\mathbf{y}^T(\tau)d\tau \geq \alpha I_{m+2}$$

for all $t \geq 0$, where $\mathbf{y} = [\mathbf{u}^T \ \ \mathbf{x}^T]^T \in \Re^{m+2}$, I_{m+2} is the $(m+2) \times (m+2)$ identity matrix, and T, α are positive constants. Then the estimation scheme (4) ensures that $\widehat{\mathbf{a}} \rightarrow \mathbf{a}$ and $\widehat{\mathbf{b}} \rightarrow \mathbf{b}$ as $t \rightarrow \infty$ for any initial $\widehat{\mathbf{a}}(0)$ and $\widehat{\mathbf{b}}(0)$.

Proof: Let $\mathbf{c} = [\mathbf{a}^T \ \ \mathbf{b}^T]^T$ and $\widehat{\mathbf{c}} = [\widehat{\mathbf{a}}^T \ \ \widehat{\mathbf{b}}^T]^T$, so that $F = \mathbf{c}^T\mathbf{y}$ and $\widehat{F} = \widehat{\mathbf{c}}^T\mathbf{y}$. Consider the Lyapunov function candidate $V_1 = \phi^T\phi/2$, where $\phi = \mathbf{c} - \widehat{\mathbf{c}}$. Differentiating V_1 along the trajectories of (4) yields, after simplification, $\dot{V}_1 = -\gamma\phi^T\mathbf{y}\mathbf{y}^T\phi$. Standard arguments from differential equation stability theory then show that, if the condition on $\mathbf{u}$ given in the statement of the lemma is satisfied, then $\phi \rightarrow \mathbf{0}$ as $t \rightarrow \infty$ and thus the desired estimates for $\mathbf{a}, \mathbf{b}$ are obtained [15]. □

Using the preceding result, it is straightforward to obtain accurate estimates for $\mathbf{a}$ and $\mathbf{b}$; for example, it is relatively easy to ensure that the condition on $\mathbf{y}$ stated in Lemma 3 is satisfied during a "training" period for the algorithm. Thus it what follows it is assumed that $\mathbf{a}$ and $\mathbf{b}$ have been identified in this manner, so that the problem of modeling the dynamical system (2) using a RNN reduces to the problem of approximating (2a) using such a network. This approximation process involves selecting the RNN weights W and functions $\mathbf{z}$ in such a way that the response of the RNN closely models the response of (2a) if each system is subjected to the same inputs. Lemma 1 indicates that an appropriate choice for $\mathbf{z}$ can be obtained using either RHONN methods or GRBFN techniques. Determining suitable values for the elements of W is ordinarily accomplished by presenting a set of input trajectories to both the dynamical system to be modeled and the RNN, and adjusting the weighting matrix W in such a way that the discrepancy between the state trajectories for the two systems is reduced; this process is often termed "training" the RNN. In what follows, we propose a robust training procedure for the RNN given in (3).

It can be seen from Lemma 1 that, if $\mathbf{z}$ is chosen so that (3) is in RHONN or GRBFN (with sufficient interconnections p) then there exists a choice for W, say W^*, for which (3) provides a close approximation of (2a). In other words, given this choice for W, the dynamical system (2a) can be written as

$$\dot{\mathbf{x}} = -A\mathbf{x} + W^*\mathbf{z}(\mathbf{x},\mathbf{u}) + \eta(\mathbf{x},\mathbf{u}) \qquad (5)$$

where $\eta : \Re^2 \times \Re^m \rightarrow \Re^2$ quantifies the (small) modeling error. The difficulty, of course, is determining W^*. A robust approach for specifying W, and thereby training the RNN, is proposed in the following theorem.

Theorem 1: Consider the following strategy for determining W:

$$\dot{\mathbf{x}}_n = -A\mathbf{x}_n + W\mathbf{z}(\mathbf{x},\mathbf{u}) \qquad (6a)$$

$$\dot{W} = -\alpha W + \beta \mathbf{e}\mathbf{z}^T \qquad (6b)$$

where $\mathbf{e} = \mathbf{x} - \mathbf{x}_n$ and α, β are positive scalar constants. This strategy ensures that $\mathbf{e}$ and $\widetilde{W} = W^* - W$ are uniformly bounded for all model mismatches η, and that the ultimate bound on $\mathbf{e}$ is small if η is small.
Proof: The identification error dynamics can be obtained by differencing (5) and (6a):

$$\dot{\mathbf{e}} = -A\mathbf{e} + \widetilde{W}\mathbf{z} + \eta \qquad (7)$$

Consider the Lyapunov function candidate

$$V_2 = \frac{1}{2}\mathbf{e}^T\mathbf{e} + \frac{1}{2\beta}\mathrm{tr}[\widetilde{W}\widetilde{W}^T] \qquad (8)$$

Differentiating (8) along (7) and simplifying yields

$$\dot{V}_2 \leq -\frac{a_m}{2} \parallel \mathbf{e} \parallel^2 - \frac{\alpha}{2\beta} \parallel \widetilde{W} \parallel_F^2 + \frac{\eta_{max}^2}{2a_m} + \frac{\alpha \parallel W^* \parallel_F^2}{2\beta}$$

where $a_m = \min(a_i)$, η_{max} is an upper bound on $\parallel \eta \parallel$, and $\parallel \cdot \parallel_F$ is the Frobenius norm. Application of Lemma 2 then permits the conclusion that $\mathbf{e}, \widetilde{W}$ are uniformly bounded and are guaranteed to converge exponentially to a compact neighborhood of the origin in $\mathbf{e}, \widetilde{W}$ space. A bound on the ultimate size of $\parallel \mathbf{e} \parallel$ can also be obtained using Lemma 2; if β is chosen reasonably large then the size of this ultimate bound is proportional to the size of the model mismatch η_{max}, so that $\mathbf{e}$ is small if η is small. □

The results summarized in Lemma 1, Lemma 3, and Theorem 1 define a strategy for modeling the uncertain tool wear dynamical system (2) using a RNN together with an (adaptively obtained) output map. If the conditions stipulated in Lemma 1 and Theorem 1 are satisfied, the response of the state $\mathbf{x}_n$ of the RNN to a given input trajectory should provide an accurate approximation of the response of the actual system to this input trajectory. Thus this RNN could, in principle, be used to give an estimate for tool wear during machining. However, it is highly desirable to introduce a feedback mechanism into this estimation algorithm, so that the wear estimate is influenced by the evolution of the actual system. One means of accomplishing this goal is described in the next section.

3.2 Robust Observer

Let $W_c \in \Re^{n \times p}$ denote the matrix of constants obtained by training the RNN (3) using the robust identification scheme summarized in Theorem 1. If the results given in Lemma 1, Lemma 3, and Theorem 1 are used to construct the RNN model of the tool wear dynamics (2), then this dynamical system can be rewritten as

$$\dot{\mathbf{x}} = -A\mathbf{x} + W_c\mathbf{z}(\mathbf{x},\mathbf{u}) + \eta(\mathbf{x},\mathbf{u}) \qquad (9a)$$

$$F^* = \mathbf{b}^T\mathbf{x} \qquad (9b)$$

where $F^* = F - \mathbf{a}^T\mathbf{u}$. In (9), η is unknown but small and $A, W_c, \mathbf{z}, \mathbf{b}$, and F^* are known (observe, for example, that F^* is known because F can be measured, $\mathbf{u}$ is known and $\mathbf{a}$ has been identified).

Our objective is now to develop an observer which can asymptotically reconstruct the tool wear state $\mathbf{x}$ from measurements of F^* despite the system uncertainty quantified by η. Note first that the original system (2) is observable [9], so that this objective is attainable. Consider the observer

$$\dot{\mathbf{x}}_o = -A\mathbf{x}_o + W_c\mathbf{z}(\mathbf{x}_o,\mathbf{u}) + L(F^* - \mathbf{b}^T\mathbf{x}_o) \qquad (10)$$

where $\mathbf{x}_o \in \Re^2$ is the estimate of the wear state $\mathbf{x}$ and $L \in \Re^{2\times 1}$ is the observer gain matrix to be specified later. Applying (10) to (9) yields

$$\dot{\mathbf{e}} = -(A + L\mathbf{b}^T)\mathbf{e} + W_c[\mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o)] + \eta \qquad (11)$$

where $\mathbf{e} = \mathbf{x} - \mathbf{x}_o$ is the state estimation error and the dependence of $\mathbf{z}$ on $\mathbf{u}$ has been supressed for clarity. The performance characteristics of the observer (10) can be determined by studying the stability properties of (11). The following fact concerning the pair of matrices $(A, \mathbf{b})$ will be helpful in this regard.

Lemma 4: If the RNN is designed so that the diagonal elements of A are not equal (this is always possible) then the eigenvalues of $(A + L\mathbf{b}^T)$ can be arbitrarily assigned through the specification of L.
Proof: It is trivial to verify that the matrix $[\mathbf{b}^T \;\; A^T\mathbf{b}]$ has full rank, and the assignability of the eigenvalues of $(A + L\mathbf{b}^T)$ follows directly from this [16]. □

The following theorem characterizes the performance capabilities of the observer (10)..

Theorem 2: The observer (10) ensures that the state estimation error e is uniformly bounded, and has an ultimate bound which can be reduced arbitrarily, provided that L is properly specified.
Proof: Suppose that L is chosen so that $\lambda = \lambda_{min}(A+L\mathbf{b}^T) = \lambda_{max}(A + L\mathbf{b}^T)/2$ is a large positive constant. Let $\Lambda = \text{diag}(\lambda \quad 2\lambda)$ and M^{-1} denote the associated (normalized) modal matrix, so that $\Lambda = M(A + L\mathbf{b}^T)M^{-1}$. Then defining $\mathbf{q} = M\mathbf{e}$ permits (11) to be rewritten

$$\dot{\mathbf{q}} = -\Lambda\mathbf{q} + MW_c[\mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o)] + M\eta \qquad (12)$$

Consider the Lyapunov function candidate $V_3 = \mathbf{q}^T\mathbf{q}/2$. Differentiating V_3 along the trajectories of (12) yields, after simplification

$$\dot{V}_3 \leq -(\lambda - K \parallel M \parallel_F \parallel W_c \parallel_F) \parallel \mathbf{q} \parallel^2 + \parallel M \parallel_F \parallel \eta \parallel \parallel \mathbf{q} \parallel$$

where $K > 0$ satisfies $K \parallel M(\mathbf{x} - \mathbf{x}_o) \parallel \geq \parallel \mathbf{z}(\mathbf{x}) - \mathbf{z}(\mathbf{x}_o) \parallel \quad \forall \mathbf{x}, \mathbf{x}_o$ (the existence of such a K is a consequence of the fact that, for RHONN and GRBFN designs, the partial derivatives of $\mathbf{z}$ are all bounded). Assume that L is chosen so that $\lambda_0 = \lambda - K \parallel M \parallel_F \parallel W_c \parallel_F$ is positive (this is always possible). Then routine manipulation permits $\dot{V}_3$ to be upper bounded as

$$\dot{V}_3 \leq -(\lambda_0 \parallel \mathbf{q} \parallel - \eta_{max} \parallel M \parallel_F) \parallel \mathbf{q} \parallel$$

From Lemma 2 it can now be concluded that $\parallel \mathbf{q} \parallel$ is uniformly bounded and that the ultimate bound on this error is $\parallel \mathbf{q} \parallel \leq \eta_{max} \parallel M \parallel_F / \lambda_0$; note that the ultimate bound on $\parallel \mathbf{q} \parallel$ (and therefore $\parallel \mathbf{e} \parallel$) can be decreased as desired by increasing λ_0 through the specification of L. □

4. Case Study: Turning

The proposed tool wear estimation algorithm was evaluated through a case study involving a turning application. In this case study, AISI medium carbon steel cylinders having 223 BHN hardness were turned using a CNC lathe with Valenite VC55 TNMA432 uncoated cemented carbide tools [9]. Cutting force measurements were obtained for all trials, and the following cutting conditions were used throughout the case study [9]: cutting speed = 200m/min, feed = 0.15mm/rev, and depth of cut = 1.27mm.

The tool wear estimation algorithm utilized in this case study consisted of an RHONN of the form given in (3) together with the state observer (10). The RHONN structure was determined using the design procedure given in [11]. The network was trained using data generated through computer simulation using the tool wear model proposed in [10]. In each trial only two cutting experiments were simulated, so that the training data sets were quite modest. The parameter values used in the observer (10) were obtained after a few iterations using computer simulation data, and no attempt was made to "tune" the parameters to achieve optimal performance. Instead, two sets of observer parameters were used: a "low gain" observer which resulted in a moderate magnitude for λ, and a "high gain" observer which placed the magnitude of λ at a large value. All details concerning the implementation of the proposed estimation algorithm, including the structure of the RHONN and the parameter values for the network and the observer, are given in [17].

To evaluate the performance of the "trained" estimation algorithm, the algorithm was tested using the experimental data obtained by Park and Ulsoy in [9]. More specifically, the force measurement trajectories reported in [9] were used as inputs to the wear state estimation algorithm to generate estimated tool wear trajectories. The estimates for the flank wear components due to abrasion (w_{f_1}) and diffusion (w_{f_2}) for a typical trial with the high gain observer are shown in Figure 1; estimates obtained using the low gain observer were similar and hence are not shown. The evolution of the total tool wear w_f obtained using the high gain observer is shown in Figure 2, and it can be seen that this estimate agrees quite closely with the experimental measurements made by Park and Ulsoy [9].

5. Conclusions

This paper presents a robust strategy for estimating tool wear in metal cutting operations. The proposed estimation algorithm consists of a recurrent neural network to model the tool wear dynamics, and a state observer to estimate the tool wear from this model using measurements of cutting force. A stability analysis is performed to demonstrate that the algorithm provides uniformly bounded tool wear estimation error and that the ultimate bound on this error can be reduced arbitrarily. The proposed approach is applied to the problem of estimating tool wear in a turning application and is shown to provide wear estimates which are in close agreement with published experimental results.

6. Acknowledgments

The research described in this paper was supported in part through contracts with the Army Research Office and Sandia National Laboratories.

7. References

1. Matsushima, K. and T. Sata, "Development of Intelligent Machine Tool" *J. Faculty of Engineering*, University of Tokyo, Vol. 35, No. 3, 1980
2. Rangwala, S., S. Liang, and D. Dornfeld, "Pattern Recognition of Acoustic Emission Signals During Punch Stretching" *Mechanical Systems and Signal Processing*, Vol. 1, No. 4, 1987, pp. 321-332
3. Rangwala, S., "Machining Process Characterization and Intelligent Tool Condition Monitoring Using Acoustic Emission Signal Analysis" Ph.D. thesis, University of California, Berkeley, 1988
4. Burke, L., "An Unsupervised Neural Network Approach to Tool Wear Identification" *IIE Transactions*, Vol. 25, No. 1, 1993, pp. 16-25
5. Yao, Y. and X. Fang, "Assessment of Chip Forming Patterns with Tool Wear Progression in Machining via Neural Networks" *International Journal of Machine Tools Manufacture*, Vol. 33, No. 1, 1993, pp. 89-102
6. Koren, Y., "Flank Wear Model of Cutting Tools Using Control Theory", *ASME Journal of Engineering for Industry*, Vol. 100, No. 1, 1978, pp. 103-109
7. Danai, K. and A. Ulsoy, "A Dynamic State Model for On-Line Tool Wear Estimation in Turning", *ASME Journal of Engineering for Industry*, Vol. 109, No. 4, 1987, pp. 396-399
8. Koren, Y., T. Ko, K. Danai, and A. Ulsoy, "Flank Wear Estimation Under Varying Cutting Conditions", *ASME J. Dynamic Sys., Meas., Control*, Vol. 113, No. 2, 1991, pp. 300-307
9. Park, J. and A. Ulsoy, "On-Line Tool Wear Estimation Using Force Measurement and a Nonlinear Observer", *ASME J. Dynamic Sys., Meas., Control*, Vol. 114, No. 4, 1992, pp. 666-672
10. Park, J. and A. Ulsoy, "On-Line Flank Wear Estimation Using an Adaptive Observer and Computer Vision: Theory and Experiment", *ASME Journal of Engineering for Industry*, Vol. 115, No. 1, 1993, pp. 30-43
11. Kosmatopoulos, E., P. Ioannou, M. Christodoulou, "Identification of Nonlinear Systems Using New Dynamic Neural Network Structures" *Proc. IEEE Conference on Decision and Control*, Tucson, AZ, December 1992
12. Sanner, R. and J.-J. Slotine, "Gaussian Networks for Direct Adaptive Control" *IEEE Trans. Neural Networks*, Vol. 3, No. 6, 1992, pp. 837-863
13. Corless, M., "Guaranteed Rates of Exponential Convergence for Uncertain Systems", *Journal of Optimization Theory and Applications*, Vol. 64, No. 3, 1990, pp. 481-494
14. Colbaugh, R., H. Seraji, and K. Glass, "Adaptive Compliant Motion Control for Dexterous Manipulators", *International Journal of Robotics Research*, Vol. 14, No. 3, 1995, pp. 270-280
15. Vidyasagar, M., *Nonlinear Systems Analysis*, 2nd Ed., Prentice Hall, Englewood Cliffs, NJ, 1993
16. Sontag, E., *Mathematical Control Theory*, Springer-Verlag, New York, 1990
17. Colbaugh, R. and K. Glass, "A Robust Approach to Real-Time Tool Wear Estimation", Robotics Laboratory Report, New Mexico State University, Las Cruces, NM, December 1993

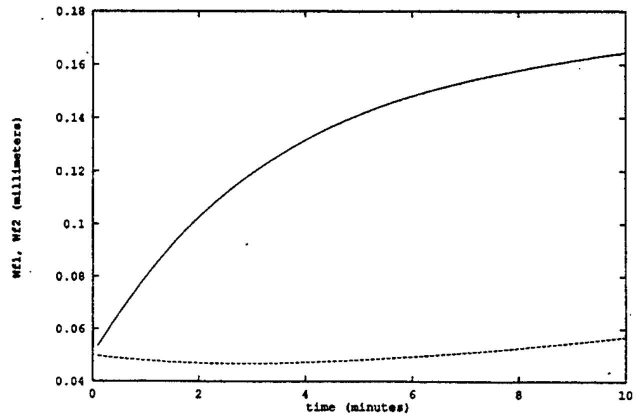


Figure 1 : Estimated flank wear components (high gain observer)

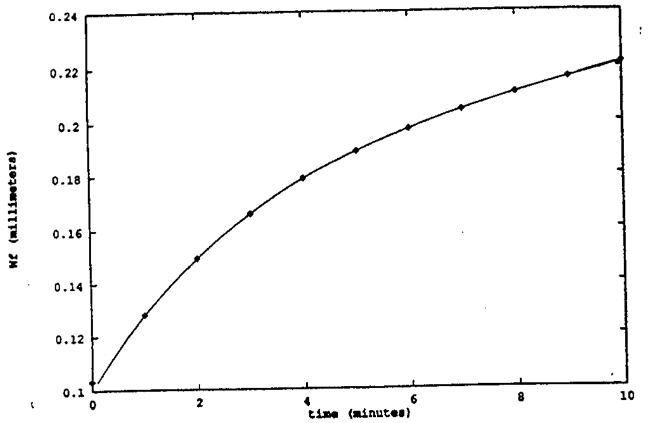


Figure 2: Estimated and measured flank wear (high gain observer)

International Journal of Robotics and Automation, Vol. 10, No. 4, 1995

STABILITY ANALYSIS FOR A CLASS OF NEURAL NETWORKS

R. Colbaugh*

Abstract

This paper considers the problem of characterizing the stability properties of the equilibria of an important class of recurrent neural networks. Sufficient conditions are given under which the neural network possesses a unique globally asymptotically stable equilibrium point for each external input. These conditions are less restrictive than those previously obtained and are easily checked, so that incorporating them in existing neural network design procedures should increase the flexibility and reduce the complexity of this synthesis process. Results are provided for both continuous-time and discrete-time networks.

Key Words

Stability analysis, recurrent neural networks, automation

1. Introduction

A recurrent neural network is a dynamical system composed of a large number of highly interconnected neuron-like subsystems. The structure of these networks leads naturally to distributed information processing and an inherent potential for efficient parallel computation. As a consequence, neural networks have attracted considerable interest as candidates for robust, high-performance computational systems. Recognizing the potential advantages of these systems for solving complex problems in real time, many researchers have studied the application of neural networks to a wide variety of automation problems. For example, neural networks have been implemented to solve problems in optimization (e.g., [1–4]), control theory (e.g., [5–7]), robotics (e.g., [8–10]), and pattern recognition (e.g., [11–13]).

One of the most popular and useful classes of neural networks proposed to date is the *additive* or *Hopfield-type* network [1, 2]. Fundamental to the successful implementation of this class of neural network as a solution to automation problems is an understanding of the qualitative properties of the system dynamics. Of particular interest for the design and utilization of these neural networks is a characterization of the network equilibria, including the relationship between the external inputs and the resulting equilibria and the stability properties of these equilibria. In many applications it is desirable, or even essential, to have a unique globally asymptotically stable equilibrium corresponding to each external input. Thus in these cases it is of great interest to the designer to have conditions on the neural network parameters that ensure that this situation is realized.

Understanding the importance of this problem, many researchers have studied the qualitative properties of the dynamics of additive neural networks (e.g., [1, 2, 14–21]). Much of the early work in this area imposed restrictive conditions on the network components and architecture to ensure the existence of asymptotically stable equilibria. For example, a common assumption was that the network possessed a symmetric connectivity between the neurons, and that the neuron input/output "activation" functions were smooth (e.g., [1, 2, 14, 21]). These conditions have been found to impose serious limitations on the capabilities of neural networks, and as a consequence recent investigations have sought to relax these restrictions while still ensuring the existence of asymptotically stable equilibria [15–20]. Most of this work has focused on local stability, so that conditions for global asymptotic stability of an equilibrium that are not restrictive and that can be easily checked are still lacking. Additionally, most of the research on the qualitative dynamics of neural networks has considered the continuous-time case, despite the fact that discrete-time networks have proven to be quite useful in applications.

This paper considers the problem of characterizing the stability properties of the equilibria of additive neural networks. Sufficient conditions are given under which the neural network possesses a unique globally asymptotically stable equilibrium point for each external input. These conditions are simple and easily checked, and are less restrictive than those obtained in other investigations. For example, it is not assumed that the connectivity between the neurons is symmetric, and the neuron activation functions are required only to be locally Lipschitz. The focus of the work is on continuous-time neural networks; however, results are also provided that ensure the analogous behaviour in the discrete-time case.

2. Preliminaries

The objective of this paper is to investigate the stability properties of additive neural networks. A continuous-time realization of this class of networks typically takes the form:

$$\dot{\mathbf{x}} = -A\mathbf{x} + W\mathbf{g}(\mathbf{x}) + \mathbf{I} \tag{1}$$

where $\mathbf{x} \in \Re^n$ is the vector of neuron activation states, $A = \mathrm{diag}[a_i] \in \Re^{n \times n}$ is a matrix modelling the self-

* Department of Mechanical Engineering, New Mexico State University, Las Cruces, NM 88003 USA (e-mail: colbaugh@nmsu.edu)

(paper no. 93-057)

inhibitions of the neurons (so that the a_i are positive constants), $W \in \Re^{n\times n}$ is a constant matrix whose elements w_{ij} define the interconnection strengths between the neurons, $\mathbf{g} = [g_1(x_1), g_2(x_2), ..., g_n(x_n)]^T : \Re^n \rightarrow \Re^n$ is a locally Lipschitz mapping representing the input/output activation functions of the neurons, and $\mathbf{I} \in \Re^n$ is the vector of external inputs to the network. In what follows it will be assumed that the functions g_i are bounded and strictly increasing, and that the solutions of (1) are defined for all time and exhibit continuous dependence on initial conditions.

In discrete time, the additive neural network model is ordinarily taken to be:

$$\mathbf{x}_{k+1} = W\mathbf{g}(\mathbf{x}_k) + \mathbf{I} \tag{2}$$

where all terms are defined as in the continuous-time model. This representation is usually referred to as a synchronous discrete-time network, and we note that this version possesses implementation advantages over models that switch asynchronously [19].

In what follows, we shall be concerned with the stability properties of the equilibria of the neural network models (1),(2). Therefore, an important component of the analysis consists in establishing the existence of an equilibrium solution for a nonlinear differential or difference equation. The next two lemmas will be helpful in this regard.

Lemma 1: Consider the continuous-time dynamical system:

$$\dot{\mathbf{x}} = B\mathbf{x} + \mathbf{f}(\mathbf{x}) \tag{3}$$

where $B \in \Re^{n\times n}$ is constant and nonsingular and $\mathbf{f} : \Re^n \rightarrow \Re^n$ is locally Lipschitz. If there exists a constant $K > 0$ such that $\| \mathbf{f}(\mathbf{x}) \| \leq K \quad \forall \mathbf{x}$ then (3) has an equilibrium solution $\mathbf{x}_{eq}$ (i.e., $\mathbf{x}_{eq}$ satisfies $B\mathbf{x}_{eq} + \mathbf{f}(\mathbf{x}_{eq}) = 0$).

Proof: We seek a solution to $B\mathbf{x} + \mathbf{f}(\mathbf{x}) = 0$ or, equivalently, to the fixed-point problem:

$$\mathbf{x} = -B^{-1}\mathbf{f}(\mathbf{x}) \tag{4}$$

Now the boundedness of $\mathbf{f}$ and invertibility of B ensures the existence of a constant $R > 0$ such that $\| B^{-1}\mathbf{f}(\mathbf{x}) \| \leq R \quad \forall \mathbf{x}$. Thus $B^{-1}\mathbf{f}(\mathbf{x})$ maps the ball $B_R(0)$ (of radius R centred at the origin) into itself, and the Brouwer fixed-point theorem then ensures that $B^{-1}\mathbf{f}(\mathbf{x})$ has a fixed point in $B_R(0)$ [22]. This fixed point is an equilibrium solution for (1). QED

Lemma 2: Consider the discrete-time dynamical system:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) \tag{5}$$

where $\mathbf{f} : \Re^n \rightarrow \Re^n$ is continuous. If there exists a constant $R > 0$ such that $\| \mathbf{f}(\mathbf{x}) \| \leq R \quad \forall \mathbf{x}$ then (5) has an equilibrium solution $\mathbf{x}_{eq}$ (i.e., $\mathbf{x}_{eq}$ satisfies $\mathbf{x}_{eq} = \mathbf{f}(\mathbf{x}_{eq})$).

Proof: Here we have immediately a fixed-point problem, so that the boundedness of $\mathbf{f}(\mathbf{x})$ and the Brouwer fixed-point theorem again ensure the existence of a fixed-point (in $B_R(0)$) and therefore an equilibrium solution for (5). QED

Our approach to analyzing the stability of the equilibrium solutions to the dynamical systems (1) and (2) is to utilize Lyapunov's direct method [22]. The following two lemmas summarize certain facts regarding this method of stability analysis and will be of direct relevance in our development. The first lemma recalls a useful result concerning the stability of continuous-time dynamical systems.

Lemma 3: Consider the continuous-time dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, where $\mathbf{f} : \Re^n \rightarrow \Re^n$ is locally Lipschitz and such that $\mathbf{f}(0) = \mathbf{0}$. If there exists a continuously differentiable function $V : \Re^n \rightarrow \Re$ that is positive-definite, radially unbounded, and such that the Lie derivative $\dot{V} = (\partial V/\partial \mathbf{x})\mathbf{f}$ is negative-definite, then the equilibrium solution $\mathbf{x} = \mathbf{0}$ is globally asymptotically stable.

Proof: The proof can be found in, among other sources, [23]. QED

The next lemma provides the analogous result for discrete-time dynamical systems.

Lemma 4: Consider the discrete-time dynamical system $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$, where $\mathbf{f} : \Re^n \rightarrow \Re^n$ is continuous and such that $\mathbf{f}(0) = \mathbf{0}$. If there exists a continuous function $V : \Re^n \rightarrow \Re$ that is positive-definite, radially unbounded, and such that the forward difference $\Delta V_k = V(\mathbf{x}_{k+1}) - V(\mathbf{x}_k)$ is negative-definite, then the equilibrium solution $\mathbf{x} = \mathbf{0}$ is globally asymptotically stable.

Proof: The proof can be found in, among other sources, [23]. QED

3. Continuous-Time Neural Networks

In this section we investigate the stability properties of the continuous-time neural network (1). The purpose of this study is to present sufficient conditions under which the network possesses a unique globally asymptotically stable equilibrium solution for each external input $\mathbf{I}$. Our main results are summarized in the following two theorems.

Theorem 1: The neural network (1) possesses a unique globally asymptotically stable equilibrium solution for each $\mathbf{I}$ provided that the matrix $W + W^T$ is negative semidefinite.

Proof: The boundedness of $\mathbf{g}$ and invertibility of A together with Lemma 1 permit the conclusion that (1) has an equilibrium solution $\mathbf{x}_{eq}$. Introducing the new coordinates $\mathbf{q} = \mathbf{x} - \mathbf{x}_{eq}$ and defining $\mathbf{f}(\mathbf{q}) = [f_1(q_1), ..., f_n(q_n)]^T = \mathbf{g}(\mathbf{q} + \mathbf{x}_{eq}) - \mathbf{g}(\mathbf{x}_{eq})$ allows (1) to be written as:

$$\dot{\mathbf{q}} = -A\mathbf{q} + W\mathbf{f}(\mathbf{q}) \tag{6}$$

where $\mathbf{f}(0) = \mathbf{0}$ (so that $\mathbf{q} = \mathbf{0}$ is an equilibrium solution for (6)) and the f_i are locally Lipschitz, bounded, and strictly increasing (because the g_i are).

Consider now the Lyapunov function candidate

$$V(\mathbf{q}) = \sum_{i=1}^{n} \int_0^{q_i} f_i(y)dy \tag{7}$$

and observe that the characteristics of $\mathbf{f}$ ensure that V is continuously differentiable, positive-definite, and radially unbounded. Computing the derivative of (7) along (6) and simplifying yields:

$$\dot{V} = -\mathbf{f}^T A\mathbf{q} + \frac{1}{2}\mathbf{f}^T(W + W^T)\mathbf{f}$$
$$\leq -\min(a_i) \parallel \mathbf{f} \parallel\parallel \mathbf{q} \parallel + \frac{1}{2}\lambda_{max}(W + W^T)\parallel \mathbf{f} \parallel^2$$

The negative-semidefiniteness of $W + W^T$ implies that $\lambda_{max}(W + W^T) \leq 0$, so that $\dot{V}$ is negative-definite and the conclusions of the theorem then follow from Lemma 3. QED

The condition that all of the eigenvalues of the matrix $W + W^T$ must be nonpositive can be relaxed if there is some additional information available regarding the functions $g_i(x_i)$; this result is the subject of the next theorem.

Theorem 2: Consider the continuous-time neural network (1), rewritten as in (6). Suppose that, in addition to the previous conditions on $\mathbf{f}(\mathbf{q})$, it is known that the functions $f_i(q_i)$ satisfy $|f_i(q_i)| \leq c_i|q_i|$ for some positive scalar constants c_i. Let $a_{min} = \min(a_i)$ and $c_{max} = \max(c_i)$. If the matrix W is such that $\lambda_{max}(W + W^T) < 2a_{min}/c_{max}$ then (6) (and therefore (1)) possesses a unique globally asymptotically stable equilibrium solution for each **I**.

Proof: Differentiating the Lyapunov function candidate (7) along the solution trajectories of (6) and simplifying as in the proof of Theorem 1 gives the following upper bound on $\dot{V}$:

$$\dot{V} \leq -a_{min} \parallel \mathbf{q} \parallel\parallel \mathbf{f} \parallel + \frac{1}{2}\lambda_{max}(W + W^T) \parallel \mathbf{f} \parallel^2$$
$$\leq -\left(\frac{a_{min}}{c_{max}} - \frac{1}{2}\lambda_{max}(W + W^T)\right) \parallel \mathbf{f} \parallel^2$$

Thus $\dot{V}$ is negative-definite if the condition on W stated in the theorem is satisfied, and the conclusions of the theorem then follow from Lemma 3. QED

Several observations can be made concerning the results summarized in Theorems 1 and 2. First, note that the global nature of the stability conclusions implies that there is a unique equilibrium solution for each external input **I**; indeed, it is easy to see that the map $\mathbf{F} : \Re^n \rightarrow \Re^n$ taking input vectors **I** to equilibria $\mathbf{x}_{eq}$ is a bijection. This latter result is important for many applications. Next, observe that the conditions on the neural network parameters and architecture that guarantee global convergence to unique equilibria are less restrictive than those obtained previously (e.g., [1, 2, 14–21]). For example, it is not assumed that the interconnection matrix W is symmetric or that the activation function **g** is differentiable. Additionally, it can be seen that these conditions are easily checked. These latter features can be useful for increasing the flexibility and reducing the complexity of the neural network design process.

4. Discrete-Time Neural Networks

In this section we study the stability properties of the equilibria of the discrete-time neural network (2). As in the previous section, the objective is to present sufficient conditions for the existence of a unique globally asymptotically stable equilibrium solution for each external input **I**. Observe first that from the boundedness of **g** and Lemma 2 it can be concluded that (2) has at least one equilibrium solution $\mathbf{x}_{eq}$ for each **I**. Utilizing the coordinate **q** and the function $\mathbf{f}(\mathbf{q})$ introduced in the previous section permits (2) to be written in the following form:

$$\mathbf{q}_{k+1} = W\mathbf{f}(\mathbf{q}_k) \qquad (8)$$

where again $\mathbf{f}(0) = 0$ so that $\mathbf{q} = 0$ is an equilibrium solution.

An immediate observation is that if $W\mathbf{f}$ is a contraction mapping, so that $\parallel W\mathbf{f}(\mathbf{q}_1) - W\mathbf{f}(\mathbf{q}_2) \parallel \leq k_1 \cdot \parallel \mathbf{q}_1 - \mathbf{q}_2 \parallel$ for some positive constant $k_1 < 1$ and for all $\mathbf{q}_1, \mathbf{q}_2$, then the origin is the unique globally asymptotically stable equilibrium solution corresponding to the given input **I**. This situation occurs, for example, if **f** is globally Lipschitz with Lipschitz constant $k_2 > 0$ and the maximum singular value σ_{max} of W satisfies $\sigma_{max} < 1/k_2$. Although this condition is easily checked, it is too restrictive for many applications. An alternative condition can be established using Lyapunov's direct method; it is the content of the next theorem.

Theorem 3: Consider the discrete-time neural network (8) (or, equivalently, (2)) and suppose that the f_i satisfy $|f_i(q_i)| \leq c_i|q_i|$ for some scalar constants $c_i > 0$. Define $C = I_n - D$, where I_n is the $n \times n$ identity matrix and the elements d_{ij} of D are given by $d_{ij} = |w_{ij}|c_j$. If the diagonal elements of W and C are nonzero and positive, respectively, and the successive principle minors of C are all positive, then the origin is a globally asymptotically stable equilibrium solution of (8).

Proof: Define a vector of positive scalar constants $\lambda \in \Re^n$ and denote by $|\mathbf{q}|$ the vector of absolute values of elements of **q**, that is, $|\mathbf{q}| = [|q_1|, ..., |q_n|]^T$. Consider the Lyapunov function candidate:

$$V(\mathbf{q}_k) = \lambda^T|\mathbf{q}_k| \qquad (9)$$

and note that V is continuous, positive-definite, and radially unbounded. Computing the forward difference of V along the solutions of (8) yields, after simplification:

and note that V is continuous, positive-definite, and radially unbounded. Computing the forward difference of V along the solutions of (8) yields, after simplification:

$$\Delta V_k = \lambda^T(|W\mathbf{f}(\mathbf{q}_k)| - |\mathbf{q}_k|)$$
$$\leq \lambda^T(|W||[\mathbf{c}\mathbf{q}_k]| - |\mathbf{q}_k|)$$
$$= -\lambda^T C|\mathbf{q}_k|$$

where $\mathbf{c} = [c_1, ..., c_n]^T$ and the notation $[\mathbf{uw}] = [u_1w_1, ..., u_nw_n]^T \in \Re^n$ is introduced. The conditions on the matrix C given in the statement of the theorem are sufficient to ensure that the row-vector $\lambda^T C$ had all positive elements [e.g., 24]. Thus ΔV_k is negative-definite and Lemma 4 implies that the origin is a globally asymptotically stable equilibrium solution for (8). QED

The observations made regarding the stability results for continuous-time neural networks apply in the discrete-time case as well. For example, it can again be seen that the map $\mathbf{F} : \Re^n \to \Re^n$ taking input vectors $\mathbf{I}$ to equilibria $\mathbf{x}_{eq}$ is bijective, which is useful for many applications. Additionally, the conditions on the neural network parameters and architecture that guarantee global convergence to unique equilibria are not restrictive and can be checked efficiently, which can reduce the difficulty of the neural network synthesis process.

5. Conclusion

This paper presents sufficient conditions under which additive neural networks possess a unique globally asymptotically stable equilibrium point for each external input. These conditions are simple and easily checked, and are less restrictive than those obtained in other studies. For example, it is not assumed that the connectivity between the neurons is symmetric or that the neuron activation functions are differentiable. Both continuous-time and discrete-time implementations of the additive neural network model are considered. Extensive computer simulations have been conducted to verify that the conditions do indeed ensure convergence of the network to the desired equilibrium; these simulations are reported in [25]. Future research will focus on the application of recurrent neural networks to the problem of estimating tool wear in machining operations [26].

Acknowledgements

The research described in this paper was supported in part through contracts with the Army Research Office (Research Triangle Park, NC) and the Jet Propulsion Laboratory (Pasadena, CA). The author gratefully acknowledges many stimulating discussions on aspects of this work with Ernest Barany and Ross Staffeldt of New Mexico State University.

References

[1] J. Hopfield & D. Tank, Neural computations of decisions in optimization problems, *Biological Cybernetics, 52*, 1985, 141–152.

[2] D. Tank & J. Hopfield, Simple "neural" optimization networks: An A/D coverter, signal decision network, and a linear programming network, *IEEE Trans. on Circuits and Systems, 33*, 1986, 533–541.

[3] M. Kennedy & L. Chua, Neural networks for nonlinear programming, *IEEE Trans. on Circuits and Systems, 35*, 1988, 554–562.

[4] A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Ruedo, J. Huertas, & E. Sanchez-Sinencio, Nonlinear switched-capacitor "neural" networks for optimization problems, *IEEE Trans. on Circuits and Systems, 37*, 1990, 384–398.

[5] R. Sanner & J.-J. Slotine, Gaussian networks for direct adaptive control, *IEEE Trans. on Neural Networks, 3*(6), 1992, 837–863.

[6] E. Sontag, Feedback stabilization using two-hidden-layer nets, *IEEE Trans. on Neural Networks, 3*(6), 1992, 981–990.

[7] L. Jin, P. Nikiforuk, & M. Gupta, Direct adaptive output tracking control using multilayered neural networks, *IEE Proc.-D, 140*(6), 1993, 393–398.

[8] J. Barhen, S. Gulati, & M. Zak, Neural learning of constrained nonlinear transformations, *Computer*, June 1989, 67–76.

[9] T. Martinetz, H. Ritter, & K. Schulten, Three-dimensional neural net for learning visuomotor coordination of a robot arm, *IEEE Trans. on Neural Networks, 1*(1), 1990, 131–136.

[10] D. DeMers & K. Kreutz-Delgado, Learning global direct inverse kinematics, *Advances in neural information processing systems 4* (San Mateo, CA: Morgan Kaufmann, 1992), 589–594.

[11] S. Grossberg, Adaptive pattern classification and universal recoding 1: Parallel development and coding of neural feature detectors, *Biological Cybernetics, 23*, 1976, 121–134.

[12] S. Rangwala, S. Liang, & D. Dornfeld, Pattern recognition of acoustic emission signals during punch stretching, *Mechanical Systems and Signal Processing, 1*(4), 1987, 321–332.

[13] L. Burke, An unsupervised neural network approach to tool wear identification, *IIE Transactions, 25*(1), 1993, 16–25.

[14] M. Cohen, & S. Grossberg, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, *IEEE Trans. on Systems, Man, and Cybernetics, 13*(5), 1983, 815–825.

[15] A. Guez, V. Protopopsecu, & J. Barhen, On the stability, storage capacity, and design of nonlinear continuous neural networks, *IEEE Trans. on Systems, Man, and Cybernetics, 18*(1), 1988, 80–87.

[16] A. Michel, J. Farrell, & W. Porod, Qualitative analysis of neural networks, *IEEE Trans. on Circuits and Systems, 36*(2), 1989, 229–243.

[17] M. Hirsch, Convergent activation dynamics in continuous time networks, *Neural Networks, 2*, 1989, 331–349.

[18] S. Aiyer, M. Niranjan, & F. Fallside, A theoretical investigation into the performance of the Hopfield model, *IEEE Trans. on Neural Networks, 1*(2), 1990, 204–215.

[19] A. Michel, J. Farrell, & H. Sun, Analysis and synthesis techniques for Hopfield type synchronous discrete time neural networks with application to associative memory, *IEEE Trans. on Circuits and Systems, 37*(11), 1990, 1356–1366.

[20] S. Sudharsanan & M. Sundareshan, Equilibrium characterization of dynamic neural networks and a systematic synthesis procedure for associative memories, *IEEE Trans. on Neural Networks, 2*(5), 1991, 509–521.

[21] M. Forti, S. Manetti, & M. Marini, A condition for global convergence of a class of symmetric neural circuits, *IEEE Trans. on Circuits and Systems, 39*(6), 1992, 480–483.

[22] J. Cronin, *Differential equations*, 2nd ed. (New York: Marcel Dekker, 1994).

[23] M. Vidyasagar, *Nonlinear systems analysis*, 2nd ed. (Englewood Cliffs, NJ: Prentice Hall, 1993).

[24] A. Michel & R. Miller, *Qualitative analysis of large scale dynamical systems* (London: Academic Press, 1977).

[25] R. Colbaugh, *Stability analysis for a class of neural networks*, Robotics Laboratory report, New Mexico State University, Las Cruces, NM, December 1993.

[26] R. Colbaugh & K. Glass, A robust approach to real-time tool wear estimation, forthcoming in *Int. J. Robotics and Automation.*

TASK 3.2

PARTICIPATING SCIENTIFIC PERSONNEL

TASK 3.2 PARTICIPATING SCIENTIFIC PERSONNEL

NAME	STATUS
Julie L. Baca	MS Candidate
Thomas Baca	BS Awarded, WSMR[1]
Dr. R. D. Colbaugh	Associate Professor
Sharlene Flachs	MS Awarded, GMC[2] Engineer
Dr. K. L. Glass	PhD Awarded, NMSU[3] Assistant Professor
Dr. Peter Pittman	PhD Awarded, LANL[4] Technical Staff
Dr. K. J. Wedeward	PhD Awarded, USNA[5] Assistant Professor

[1] White Sands Missile Range
[2] General Motors Corporation
[3] New Mexico State University
[4] Los Alamos National Laboratory
[5] United States Naval Academy

CONTACT STRESS AND RELATED THERMAL/MECHANICAL MODELING

George Mulholland
Joseph Genin

Department of Mechanical Engineering
New Mexico State University

Final Report for Task 3.3 of Applied Manufacturing Research
Submitted to the Army Research Office

February 1996

TASK 3.3

PUBLICATIONS AND TECHNICAL REPORTS PUBLISHED

TASK 3.3

PARTICIPATING SCIENTIFIC PERSONNEL

TASK 3.3 PARTICIPATING SCIENTIFIC PERSONNEL

NAME	STATUS
Guiguan Chen	MS Awarded and PhD Candidate
Qiwei Dong	MS Awarded
Dr. J. Genin	Professor
Kristi Keffer	MS Awarded
Robert I. McElroy	PhD Candidate
Muktheswara Meka	MS Candidate
Dr. G. P. Mulholland	Professor, Associate Dean
Mahesh K. Samineni	MS Awarded
Suresh G. Shivnani	MS Awarded
Kamal F. Shouman	MS Awarded
Valentina Tscharnotskaia	BS, MS Awarded
Dr. W. Xu	Assistant Professor
Xuan Xu	MS Awarded
Xiaoming Yang	PhD Candidate

MATERIAL-CUTTING TOOL INTERFACES AND ION IMPLANTATION

Leon Cox
James Bentley
Brian Lambert

Department of Mechanical Engineering
New Mexico State University

Final Report for Task 3.4 of Applied Manufacturing Research
Submitted to the Army Research Office

February 1996

Material-Cutting Tool Interfaces and Ion Implantation Research Task 3.4 Final Report

EXECUTIVE SUMMARY

The first project objective to determine the overall surface characteristics of the tool edge/substrate interface and the correlated determination of the chemical composition of the surfaces of that interface was experimentally driven, confined to drill bits and finally analyzed using Raman spectroscopy. This work addresses the drilling operation when special additives have been added to a petroleum based lubricant. The purpose of these special additives is to enhance the penetration of carbon bearing liquids into the area of the cutting edge event, and to aid in the generation of very hard carbon structures, resulting in new cutting parameters based on the experimental findings. Theoretical arguments have shown how the in-situ generation of hard carbon species can increase tool life by reducing operating wear. Experimental verification of the hypothesis requires the generation of measurable amounts of carbon at the tool/substrate interface. Follow-on research should focus on a method to generate a substantial amount of the carbonaceous material identified by spectroscopy as being possibly "hard carbon". A planned for journal publication by Prof. Jim Bentley and Prof. Leon Cox titled "Raman Measurements of Carbon Complexes Formed at the Tool/Workpiece Interface in a Lubricated Drilling Operation" is included.

The second objective to develop, implement and verify the results consists of work by Prof. Leon Cox and Dr. Amjed Al-Ghanim devising a methodology of collecting, storing and utilizing machining strategy data that utilized Artificial Neural Networks (ANN's). This work was initiated in task 3.1 and is included here because it was a concurrent task to absorb the results of the experimental findings. The paper published in ASME Press Intelligent Engineering Systems Through Artificial Neural Networks, titled "Machining Strategies Collected, Stored and Utilized with ANN's" is included.

The third objective to determine the feasibility of utilizing ion implanted cutting tools …

Raman Measurements of Carbon Complexes Formed at the Tool/Workpiece Interface in a Lubricated Drilling Operation

James H. Bentley and Leon Cox
New Mexico State University
Las Cruces, NM 88003

ABSTRACT

Cutting fluids have been characterized in machining operations for lubricity and/or cooling. The chemistry of cutting fluids in a drilling operation is examined by means of analyses of Raman spectra of carbonaceous materials present at the cutting edge of a high-speed steel drill bit, with a selected variety of cutting fluids and additives.

I. INTRODUCTION

At some point in product development, modern manufacturing almost always makes use of discrete machining operations, including milling, drilling, or turning. The efficacy of any machining operation is dependent upon several operational parameters, including the nature of the cutting fluid, which may be used with or without additives. This work addresses the drilling operation when special additives have been added to a petroleum based lubricant. The purpose of these special additives is to enhance the penetration of carbon bearing liquids into the area of the cutting edge event, and to aid in the generation of very hard carbon structures, resulting in new cutting parameters based on the experimental findings.

II. EXAMINATION OF THE PROBLEM

Drilling is a complex 3-dimensional cutting operation. The outer edge of the drill bit produces chips by shearing while the material under the chisel edge is subjected to severe deformation by great localized heat and pressure. In any detailed, microscopic examination of the events occurring at the cutting edge of a sharp, lubricated cutting tool, the subject rapidly turns to the field of chemistry. This is so because the events that occur at the cutting edge are at high localized pressure and temperature. Such conditions, at times with oxygen present and at times with little or no oxygen present, can act as a micro chemical reactor, and result in the disproportionation of organic lubricant molecules and the recombination of molecular fragments into products very different from the starting materials. If some of the disproportionated carbonaceous material should recombine either as adamantane carbon or any other type of structurally complex carbon which is very hard (1), the implications for machine tooling can be extremely important.

Cutting fluids are utilized extensively in machining operations as coolants and/or lubricants.

When properly applied, cutting fluids can increase productivity and reduce costs through greater speeds, feeds, and depths of cut which are the major factors determining tool-life. The purpose of cutting fluids is to increase tool life by reducing friction and heat. This is done by reducing the forces exerted by the machine tool, cooling the cutting zone, reducing distortions, washing away chips, and protecting newly machined surfaces from environmental corrosion.

The asperities between the cutting tool and workpiece are effective storage places for small quantities of cutting fluids which are then subjected to the severity of the machining operation. The mechanism of the cutting fluid penetrating to the source of the tool/workpiece interface has been shown to be capillary action, and is a function of the molecular size of the cutting fluid components, among other characteristics (4). Since molecules with high dipole moments are very good at inducing inverted (and therefore attracting) dipoles in highly polarizable materials, they should bind electrostatically to the surfaces of cutting tool/substrate events to a greater degree than hydrocarbon molecules having little or no dipole moments. This is especially true of metallic tools and substrates, since metals possess a great deal of loosely held electron density, which makes them very easy to polarize. If the additive molecules are good organic solvents to boot, they will solvate the normal lubricant molecules, resulting in still more hydrocarbon density and therefore available carbon density at the cutting edge.

Theoretical arguments have shown how the in-situ generation of hard carbon species can increase tool life by reducing operating wear. Experimental verification of the hypothesis requires the generation of measurable amounts of carbon at the tool/substrate interface.

III. EXPERIMENT

General

Standard, off-the-shelf, 118 degree, one quarter inch diameter high speed steel drill bits were chosen for experimental use and were plunged into the end of one inch diameter AISI 1020 cold rolled steel bars in a vertical orientation. The vertical orientation assured easy retention of the lubricant and its breakdown products. The process consisted of drilling in a standard deep-hole cycle manufacturing method. Deep-hole drilling in steel was used because it best fits computer numerical controlled (CNC) machining practices found in industry today. Further, the CNC machine tool was instrumented with a load cell and the forces of the plunge were recorded with a time-based plotter. It was suspected that results from the force diagrams might show a difference according to cutting fluid. The result of this expectation was inconclusive due to the inadequacies of the load cell, and the physical variability of the drilling operation. Appendix A contains the time based plots where the first force/time portion of the plot is without cutting fluid and the second is after tool withdrawal and the cutting fluid injected into the cavity. A computerized data acquisition system utilizing the National Instruments I/O board and the LabView software to capture this data was also initiated but finally could not be used because of the limitations in the software program.

Parameters for the drilling operation were calculated from the Metcut data handbooks (3: p3-5). Cutting speed was determined to be 90 feet per minute and the resulting Speed=1600 rpm. Each successive plunge depth was set for 0.15 inches, which is just over one-half the diameter of the tool. Feed rate was established at 0.006 inches per revolution, according to the handbook.

Lubricants and Additives

Lubricating oils used in machining operations normally consist of two general types of organics - paraffinic and aromatic. Most oils are a mixture of these materials. Of course many oils also have additives for special purposes, such as "clean" cutting and "smooth" machining. Since in this investigation our purpose is to discover adamantane or other very hard types of carbon, and since it is evident, a priori, that large quantities of any kind of carbon do not remain in-situ at the normal cutting tool interface, compounds have been added to cutting fluids whose effects may be to generate more such carbon than would normally be generated in ordinary metal cutting operations.

The following comments on lubricants will list the lubricant according to a number code, and make specific comments as to how that particular lubricant/additive combination may be unique in its function. All the mixes are 50/50 by volume.

The basic lubricants consisted of a mixed hydrocarbon fraction taken from 175 - 210 degrees C, light machine oil, 30 wt. Oil, and cutting oil with sulfur.

The additives were:

1. *Styrene*. An aromatic material which has considerable unsaturation. A relatively flat, ring molecule which can fit easily into small crevasses, this material can also readily polymerize into a linear polymer containing considerable aromatic unsaturation.

2. *Iodomethane*. A very simple, small molecule containing the halogen, iodine. This covalent iodine compound will possess a smaller dipole moment that its other counterparts, and the iodine atom is very large. However the iodine atom is considerably polarizable, and will respond to localized dipole moments better than the other halogens.

3. *2,4-Pentanedione*. This molecule is a strong covalent oxygen chelator for positive centers, which forms a flat structure when it chelates.

4. *Propargyl chloride*. A small molecule containing a lot of non-aromatic unsaturation and a large dipole moment developed by the allylic chlorine. This material will not polymerize under normal conditions.

5. *3-Methyl-2-butanone*. An oil thinning organic solvent containing a large dipole moment in the ketonic carbonyl.

6. *1,2-Dichlorobenzene*. An unsaturated, aromatic material with considerable molecular dipole moment in the aromatic nucleus. This is a relatively small, flat, ring molecule which will be able to penetrate into small crevasses.

7. *Sulfolane or tetramethylene sulfone*. This molecule possesses covalent sulfur in a reasonably small ring. The sulfur is in an oxidized state and has considerable dipole moment. Sulfolane is infinitely miscible with water and will not mix as well with regular lubricants as many of the other additives will. However it will act to add water to the lubricant, gleaned from atmospheric sources or from the results of the high temperature breakdown of hydrocarbons at the cutting interface.

8. *Acetonitrile*. This is a very small molecule possessing a large dipole moment in the covalent cyano group. It should act as a thinner, but may be too polar to mix well with hydrocarbon lubricants. It also possesses considerable unsaturation in the carbon/nitrogen triple bond.

9. *Tetramethylenediamine*. This molecule is a moderate sized nitrogen based non-covalent chelator for positive centers. It possesses moderate dipole moments, but should be nicely organic soluble due to the complete methylation of the amine groups.

10. *1,4-Dioxane*. A small but somewhat chair-shaped ring molecule of great organic solvent power, having two oxygens on opposite sides of the ring. The molecule possesses considerable internal dipole moments but no net molecular dipole moment.

11. *Dimethylacetamide*. A small molecule which can slip into molecular crevasses easily. This material is a powerful solvent, containing a considerable dipole moment, and is also water miscible.

12. *1,1,1-Trichloroethane*. A highly chlorinated molecule containing both large internal and external dipole moments.

13. *Dibromoethane*. A fairly small molecule with covalent bromine on adjacent carbons. This material will have dipole moments, but will be less polar and more polarizable than the corresponding chlorine compound.

14. *Chloroform*. This is a small molecule with a large dipole moment, which will enter into small crevasses and bind well with metals by inducing considerable polarization.

15. *Phenylacetylene*. A fairly small, aromatic ring molecule with considerable unsaturation This material could be a very good source of "breakdown" carbon for the proposed adamantane structure, due to the great amount of unsaturation and the great mobility of the small, flat molecule. The molecule is highly polarizable.

Additive Characteristics

Generally there are three separate individual characteristics which report to have beneficial effects for a lubricant additive, with respect to our stated thesis of the possible generation of hard carbon at the cutting tool edge. These are as follows.

1. *Molecular Dipole Moment*. This characteristic helps define the extent to which the molecule will generate induced dipole moments on the tool bit surface. This, in turn, will determine the

extent to which the molecule will establish dipole bonding and "wet" the tool bit surface, and cause it to be maximally present at the cutting edge of the machine tool.

2. *Molecular Physical Size/Shape.* The molecular size, or vander Waal radius, determines how well the molecule is able to pack into small tool bit crevasses, according to the size of the crevasse and their mutual steric interference. The effective "size" of a molecule is determined by the number of atoms in the structure and the size of the atoms. But it is also determined by the three dimensional shape of the molecule as well. A flat molecular structure, such as a phenyl group, will occupy less volume than the corresponding saturated structure.

3. *Degree of Unsaturation.* The degree of unsaturation determines how much the starting molecule is similar to carbon at the beginning. This simply means that, molecule for molecule, there are fewer interfering atoms to deal with before rearranging the molecule's molecular structure into that of pure carbon. The degree of unsaturation of a molecule possessing a phenyl group is three, and one possessing a phenyl group plus an additional carbon/carbon double bond (such as styrene) is four. It is important to note that this quick analysis does not address the type of carbon which may be formed - that is, the state of hybridization of the carbon atoms or the physical structures.

There is an additional characteristic to consider concerning unsaturation - that of the polymerizability of the molecule. A molecule such as styrene is strongly polymerizable, where a very similar molecule, phenyl acetylene, is not normally polymerizable. "Normally" may be the key operative word, here, since the conditions of the cutting tool edge can be considered not normal, according to polymerization technology, and the acetylene molecule may indeed be able to form a complicated, at least partially unsaturated polymer chain under the unusual conditions. The presence of a carbon backbone polymer at the cutting edge event could alter the physics of molecular surface force attachment, and the subsequent chemistry done at the cutting edge. Any effects of polymerizations will not be examined specifically.

The lubricant additives were chosen in an effort to identify the types of molecular structures which will be most efficacious in generating diamond-like or other very hard carbon structures (1) in the cutting edge event. Simple dipole containing molecules were compared with more complicated molecules, possessing more diversified and more extensive dipole structures, and in some cases possessing highly polarizable structures themselves. Dipole generators of electron withdrawing entities from simple halogens to cyano groups were compared, with carbon/carbon double and triple bonds acting as charge dipole extenders in some cases.

Spectrometric Instrumentation

Raman measurements required the drill bit samples to be spectrally clean, and necessitated cleaning by submersion and ultrasonic agitation in tetrahydrofuran, followed by spectrally pure methanol. This was followed by air drying and storage in polyethylene film.

The samples were taken to Instruments S.A. (SPEX) in Edison, N.J. and under the supervision of Howard Schaffer of SPEX they were examined using a T64000 microprobe equipped with a CCD detection system and the SpectraMax [tm][4-7] spectroscopic acquisition and analysis software.

The data from the SpectraMax was output into a generic spreadsheet format. Specific results are shown in the attached Raman graphs.

IV. RESULTS

General

Making measurements on the small amounts of reaction products generated by a drilling event proved to be very difficult. It was earlier expected that in-situ Auger electron spectroscopy or even electron diffraction of laboratory isolated reaction materials could be used to identify the carbon products of the drilling procedure. This proved impossible. The instrumentation available to the university at that time consisted of surface analysis equipment manufactured by Perkin-Elmer, located at the University of Texas at El Paso (UTEP) and the NASA WSTF facility located NE of Las Cruces. The spatial resolving power of the UTEP instrumentation was not sufficient to do analyses on drill bits. The NASA instrumentation could not differentiate between sp^3 and sp^2 carbon without an X-ray attachment which would have cost approximately $75,000 to install. Even then the differentiation would require far more carbonaceous material than we could possibly make available.

Raman Utilization

If one is trying to detect the presence of diamond-like carbon in in the presence of graphitic and other more complicated thin film carbon structures, Raman technology may be the technology of choice. The ability of microRaman (MR) spectroscopy to perform MR measurements on a surface spot size as small as about one micron is extraordinarily useful for surface measurements dealing with the very small areas defined by machine tool cutting edges. This applies specifically to the drill bit cutting edge.

Since Raman spectroscopy has been utilized for characterizing hard carbon film coatings on computer discs, we reasoned that it would work for our experiments. Further, it has been found that spectral characteristics can be quantitatively correlated with the tribological properties of the films (2). Therefore, microRaman spectroscopy was chosen for the measurements.

In our investigation, we focused the MR laser as close to the cutting edge of the drill bit as seemed reasonable, and made a number of measurements along each drill bit edge. The major area of interest was at the cutting edge at about 3/4 of the radius from the drill bit axis. We chose this distance because it was the distance from the axis where maximum heat and pressure should be generated in order to cause disproportination of lubricant molecules, and recombination of elemental carbon into the forms found in-situ.

Literature Comparisons

The very sharp resonance for classical diamond in Raman spectroscopy is located at 1332 cm-1. There are two other broad peaks that may be present in the Raman spectra of carbon films in the immediate vicinity of the diamond peak. These have been defined in the literature as the mixed double and single bonded carbon (or sp^2 and^{4}sp^8 hybridized) band and the graphitic (or sp^2

hybridized) bands, containing predominately carbon/carbon double bonds (1,5,6,7,8). Ager (11) discussed the mechanical properties of the G-band frequency and the tribochemical wear rate, the abrasive wear rate and the ratio of the area of the D- and G-bands. The graph, Fig. 1, is a reproduction of that discussion and the graph, Fig. 2, is a typical finding from the MR scans which we obtained from the drill bits.

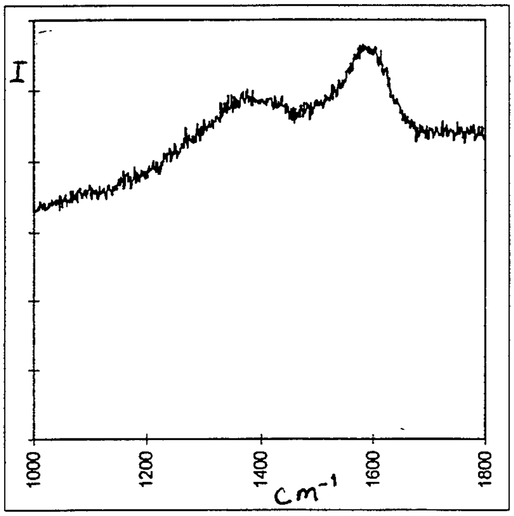


Fig. 1 Raman spectra of an a-C film (sputtered carbon) for thin film disk media, Ager (11)

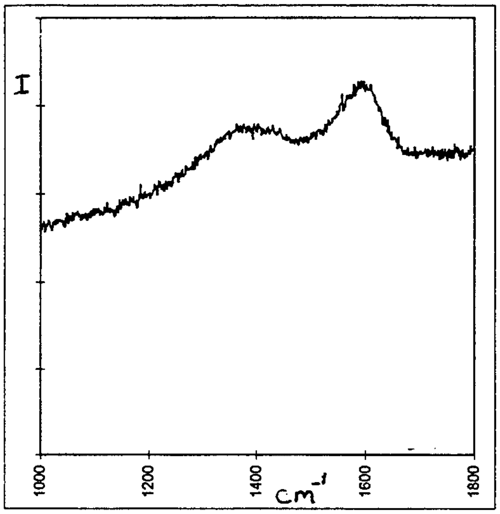


Fig. 2 Raman spectra of a drill bit after drilling with cutting oil and 1,2-Dichlorobenzene

Nuclear magnetic resonance (NMR) spectroscopy (5,6) has been used to make measurements on carbon films. Both carbon 13 magic angle spinning (5) and static single pulse excitation of the carbon 13 nucleus (6) have measured a range of sp^2 versus sp^3 hybridized carbon atoms in sputtered carbon films, although in one examination the free radical content of the films interfered greatly with the measurement (6).

A large free radical content in a carbon film indicates a large number of so called "dangling bonds", which are, per force, formed at the surfaces of pure carbon structures of both sp^2 and sp^3 hybridization. If a specimen of carbon film possesses a large number of such free radicals (along with both sp^2 and sp^3 hybridization), then it is reasonable to assume that it is composed of a large number of small individual domains of those sp^2 and sp^3 carbon atom groups. This is especially true if the hydrogen content of the film is low (6). This gives further credence to the existence of separate graphitic and adamantane carbon domains in sputtered carbon films, and due to the similarity of Raman spectra, to the case of carbon structures generated by the cutting event in machining. Specific measurements utilizing resonance Raman spectroscopy has determined the presence of sp^2 (pi bonded) carbon structures directly (5).

Appendix B contains the relevant MR scans and can be compare to Fig.s 1 & 2. Only those scans that showed desired results are included due to the large number of plots taken. The Raman spectral scans taken from our used drill bits were striking similar to those taken from sputtered carbon films used in research to extend the wear life of various surfaces (7,8,9,11,12). Many but not all of the MR measurements resulted in broad peak sets centered at about 1350 cm-1 (mixed carbon) and about 1550 to 1600 cm-1 (graphitic carbon). In all cases the graphitic band was larger than the mixed band, although in some cases they were nearly the same size. The mixed carbon peaks were about 150 cm-1 in width and the graphitic peaks about 120 cm-1 in width.

V. DISCUSSION

General

Small carbon particulates, either adamantane or graphitic in type, should respond to Raman measurements according to the 'purity' of the structures. By this we mean that if the interior of a carbon particulate is of a pure type, then the Raman measurement will respond to that medium, but with a signature modified by the Raman active (impure) nature of the surface of the particle. Logically, a range of sizes of otherwise identical particles might result in a band, or broad peak Raman signature, due to the varying effects forced by surface to volume ratios. Notably, a well defined diamond structure, typified by a gemstone diamond, results in a sharp peak, located characteristically at 1332 cm-1.

In association with thin deposited carbon films, the literature is replete with Raman scans containing two broad bands located at approximately 1550 and 1350 cm-1, which have been interpreted as respectively, a) evidence for graphitic carbon (with a great deal of sp^2 hybridized molecular structures) and, b) evidence for a complicated mixture of sp^2 and sp^3 hybridized carbon molecular structures. These interpretations are less than clear and explicit, especially in the case of the lower energy peak at 1350 cm-1. However, it seems evident that a reasonable number of the Raman active bonds in the 1350 cm-1 peaks will be sp^3 hybrids, since small fractions of such bonding would simply widen the higher energy graphite peak toward lower energies.

If one also assumes that the Raman sensitivity for the lower energy peak molecule set is roughly equivalent to that of the higher, then one can logically expect the peak areas to roughly reflect amounts of sp^2 versus sp^3 carbon/carbon bonding. (Note that this assumption is quite different from assuming the Raman sensitivity of diamond and graphite as being equal - the literature measurements indicating that graphite is some 50 times more sensitive than diamond to Raman excitation.) Such argumentation immediately leads to the argument that the 1350 cm-1 band may contain a reasonable amount of sp^2 hybridized carbon(10), although this conclusion is not unambiguously determined in the literature.

If it is further assumed that these 'very hard' carbon structures do not contain a great deal of hydrogen(7) (on the order of much less than twice the stoichiometry), then the carbon/carbon bonding should reasonably contain considerable adamantane or adamantane type carbon structures, and the presence of a certain amount of small particulate diamond must at least be considered. This would not necessarily show up at 1332 cm-1 in Raman measurements, due to the relatively low Raman response of the pure carbon adamantane structure. Such a supposition would argue that the broad 1350 cm-1 peak was a collection of small sp^2 and sp^3 hybridized carbon particulates (either covalently connected or not) possessing a relatively large surface to volume ratio, due to the very small sizes, at least some of which must have surface chemistries consisting of a fairly large number of carbon/hydrogen bonds.

Mixed Carbon Structures

The above supposition does not exclude the possibility that there also exists, in this complex of carbon structures, particulates consisting predominately of graphitic carbon, or at least carbon structures which contain large numbers of carbon/carbon double bonds. Such a complex would be almost completely carbon by elemental analysis, but might contain small domains of graphitic carbon interconnected covalently by small domains of adamantane carbon, forming (at last) a greatly extended, mostly carbon three dimensional complex, with very little ability for mutual group translational motion. Such a structure might well test as a very hard material, since molecular group motion to any extent would require the breaking of a number of strong covalent carbon/carbon bonds or extensive bending of molecular structures. These structures might look very different, spectrometrically, from gemstone diamond, but could nevertheless approach diamond in structural hardness.

It should be observed that pure graphite (at atmospheric pressure) is a soft material, and is used as a solid lubricant. But under vacuum conditions it has been found to be an abrasive. The anecdotal reason is that atmospheric gas molecules act as "roller bearings" between the large, unconnected graphite sheets, which effectively disappears in a good vacuum. This experimental observation re-inforces the theoretical argument that the graphite planes are extremely rigid in two dimensions and still quite rigid in the third dimension, due to plane-on-plane re-inforcement. If such structural planes were immobilized by extensive carbon cross-linking, even at atmospheric pressure, the result arguably could be a very hard and abrasive carbon material, possessing only a modicum of sp^3 carbon crosslinking between graphitic sheets.

VI. CONCLUSION

This result, and the fact that the literature accepts the 1500 to 1600 cm-1 band as graphitic in origin, causes us to propose that the lower energy broad peak located about 1350 cm-1 consists of 'protographite' and 'protodiamonds' - or basically adamantane carbon structures of relatively small molecular weight compounded with similar sized graphitic structures(10), each type possessing corresponding large (perhaps C-H containing) surfaces contributing to the varying Raman wavelengths defining the band. It is also postulated that these carbon/carbon double bonded structures (graphitic) and the carbon/carbon single bonded structures (adamantane) may be connected by the following possible covalent linkages: a) one or more methylene groups in series, b) direct bonding between an apical or equitorial adamantane carbon and a graphitic double bonded carbon, or c) any of a number of possible partially saturated carbocycles or linear arrangements bonded covalently to individual graphitic or adamantane domains, some of which may originate containing 'dangling bonds'.

We believe that the absence of a strong 1332 cm-1 Raman peak eliminates the possibility of adamantane structures of large extent, relative[4-11] to atomic or small molecule sizes. Due to the large size of some of the 1350 cm-1 peaks, we believe that there is at least the possibility that a very hard form of relatively pure carbon is being formed by the heat and pressure of the cutting operation, and that this relatively pure carbon may consist of very small graphitic and adamantane domains, connected covalently by sp^3 or mixed sp^3/sp^2 hybrid carbon structures. The available literature dealing with sputtered carbon films states that such carbon structures have been measured to be very hard - in the range of 20% of that of diamond (9)

Follow-on research should focus on a method to generate a substantial amount of the carbonaceous material identified by spectroscopy as being possibly 'hard carbon'. This material can then be subjected to hardness tests to finally determine whether damond-like, hard carbon is generated at the cutting edge of a tool steel drilling event.

BIBLIOGRAPHY

1. Knight and White. "Characterization of Diamond Films by Raman Spectroscopy", J. of Materials Research, Vol 4, pp385-393, Apr 1989.

2. Ager et al. "Raman and resistivity investigations of carbon overcoats of thin-film media: Correlations with tribological properties", J. Appl. Phys., Vol 69, pp5748-5750, April 1991.

3. Machining Data Handbook, 3rd Edition, pp 16-3-6, MDC, 1986.

4. Kalpakjian, Manufacturing Engineering and Technology, pp 636-640, Addison Wesley, 1989

5. Kaplan, Jansen, and Machonkin. "Characterization of Amorphous Carbon-Hydrogen Films by Solid State Nuclear Magnetic Resonance", Applied Physics Letters, Vol 47, pp750-753, Oct 1985.

6. Pan, Pruski, and Gerstein. "Local Coordination of Carbon Atoms in Amorphous Carbon", Physical Review B, Vol 44, pp6741-6745, Oct 1991.

7. Ramsteiner and Wagner. Applied Physics Letters, Vol 51, pp1355-1357, Oct 1987.

8. Dillon, Woollam, and Katkanant. "Use of Raman Scattering to Investigate Disorder and Crystallite Formation in As-Deposited and Annealed Carbon Films", Physical Review B, Vol 29, pp3482-3489, 1984.

9. Angus and Jansen. "Dense 'Diamondlike' Hydrocarbons as Random Covalent Networks", J. Vacuum Science Tech. A, Vol 6, pp1778-1782, 1988.

10. Marchon, et al. "Raman and Resistivity Investigations of Carbon Overcoats of Thin-film Media: Correlations with Tribological Properties", J. Applied Physics, Vol 69, pp5748-5750, April 1991.

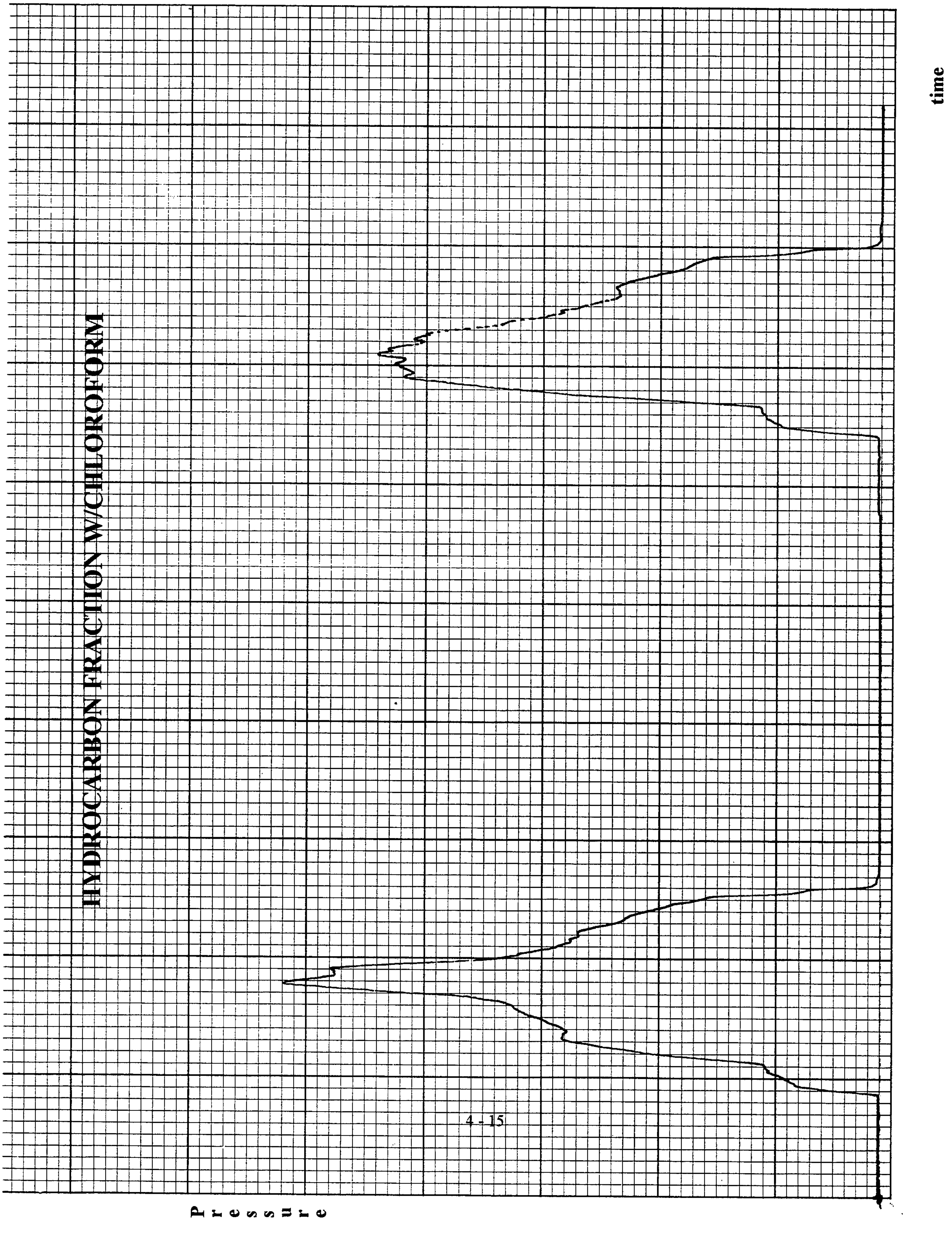
11. Ager. "Optical Characterization of Sputtered Carbon Films", IEEE Transactions on Magnetics, Vol 29, pp259-263, Jan 1993.

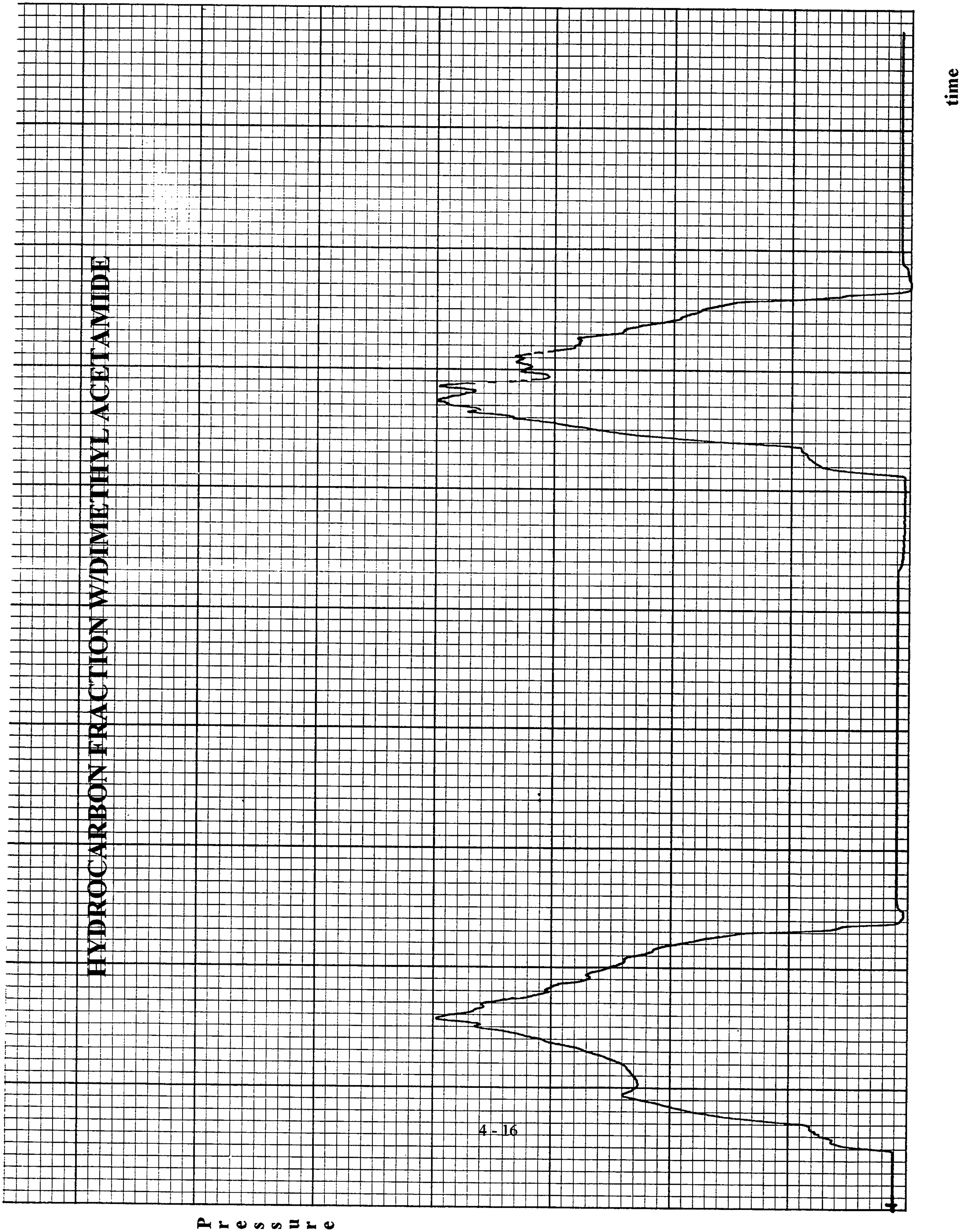
12. Robertson. "Mechanical Properties and Coordinations of Amorphous Carbons", Physical Review Letters, Vol 68, pp220-223, Jan 1992.

APPENDIX A.

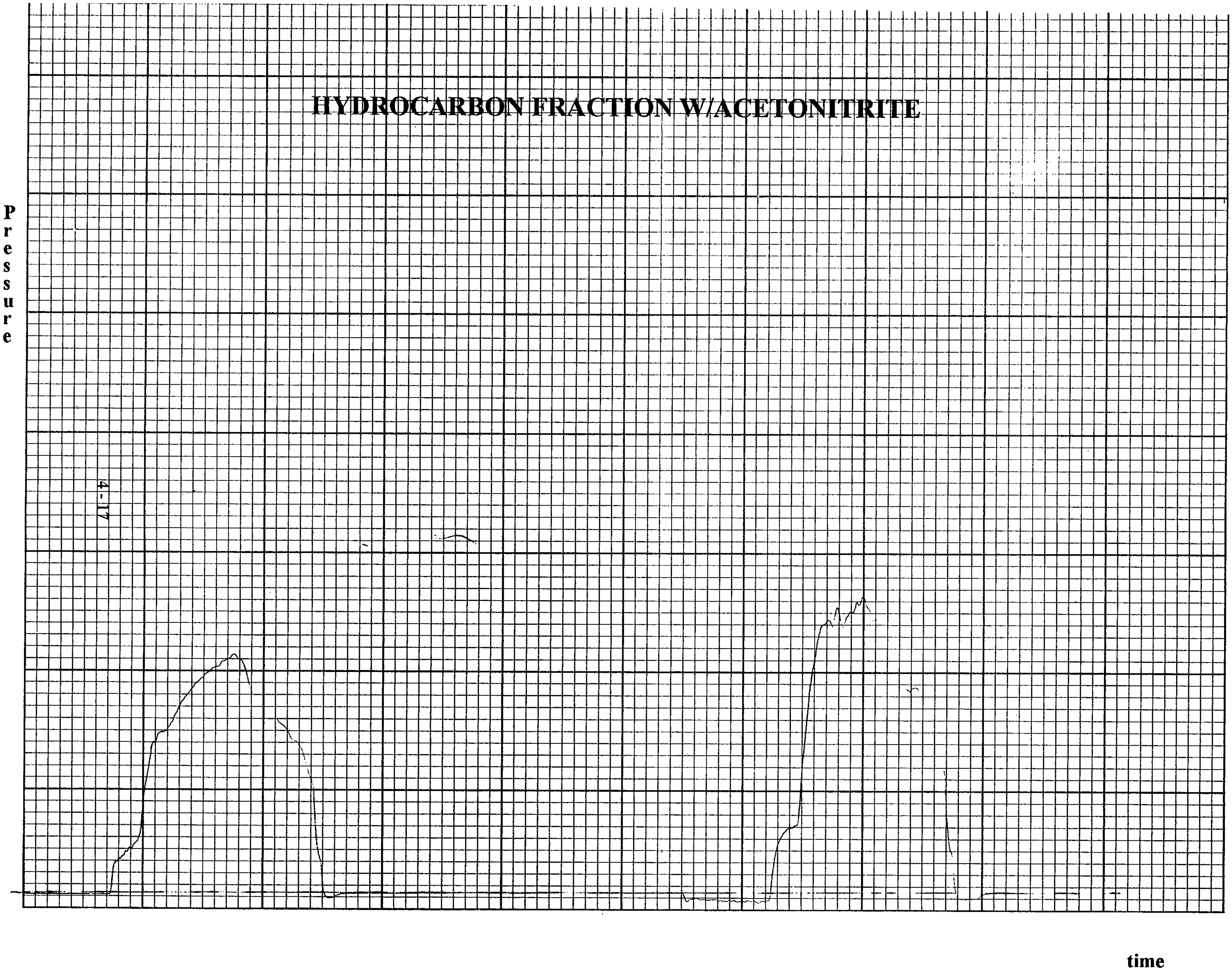
The force plots for the drilling operation corresponding to the MR scans found in Appendix B.

HYDROCARBON FRACTION W/CHLOROFORM



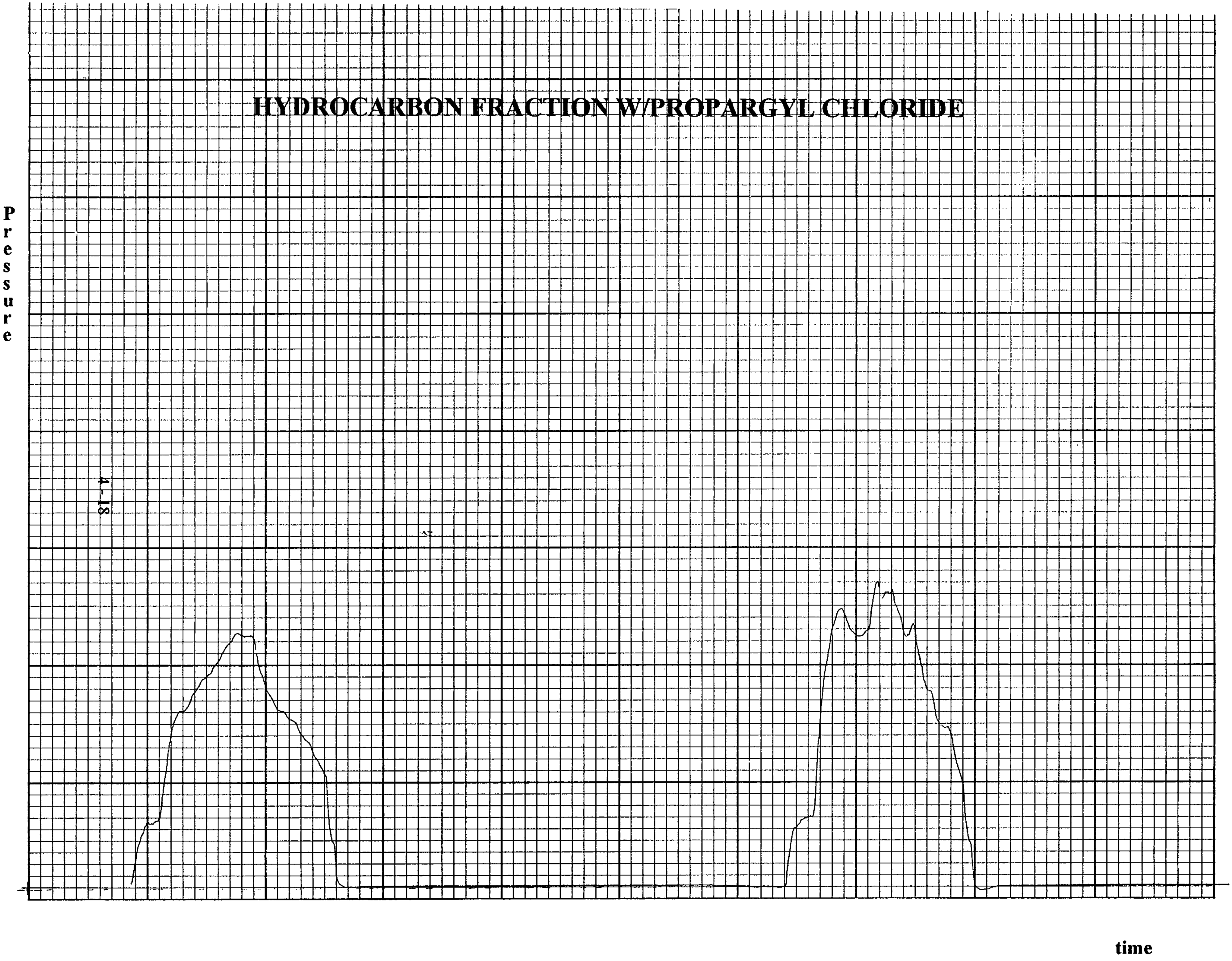


HYDROCARBON FRACTION W/DIMETHYL ACETAMIDE

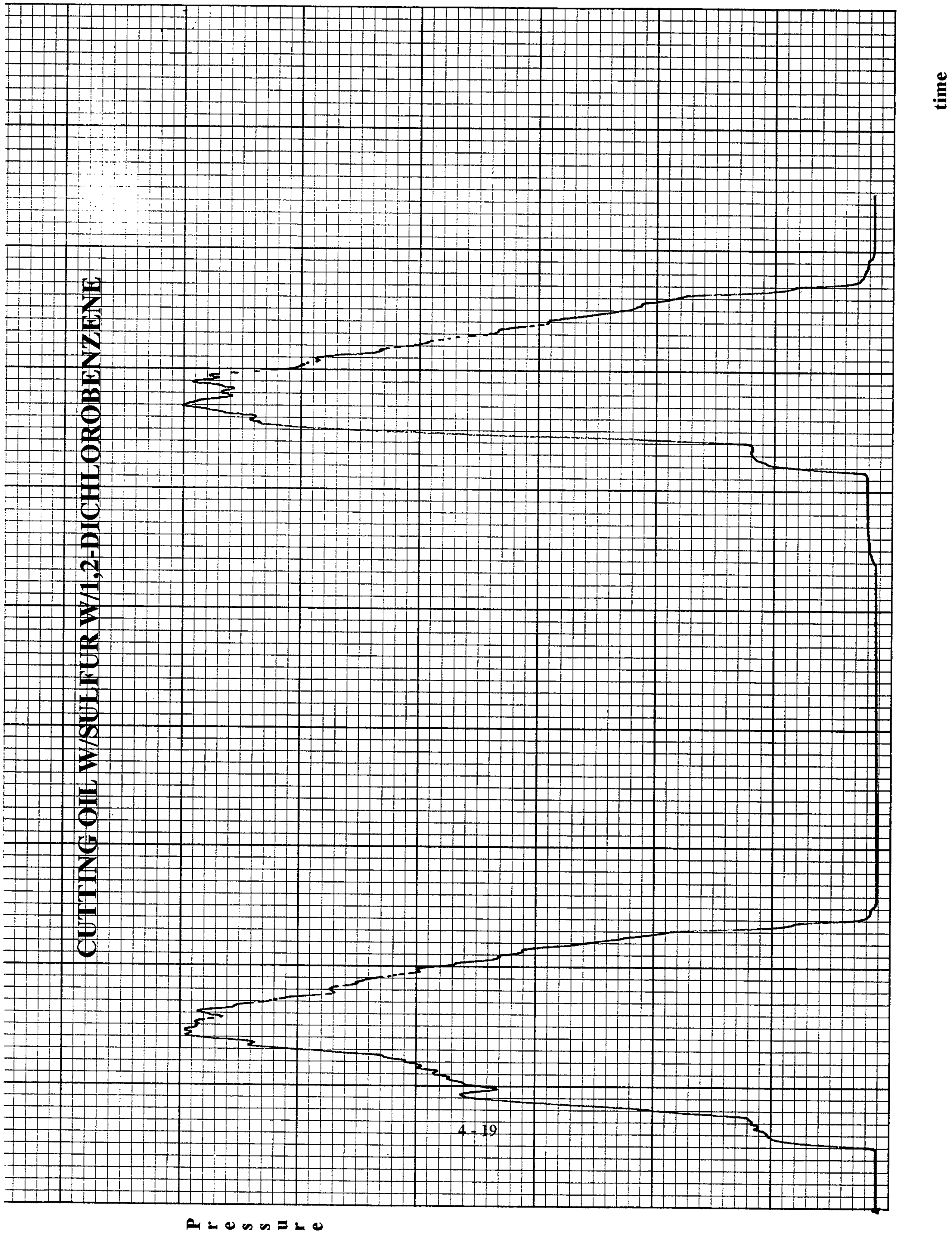


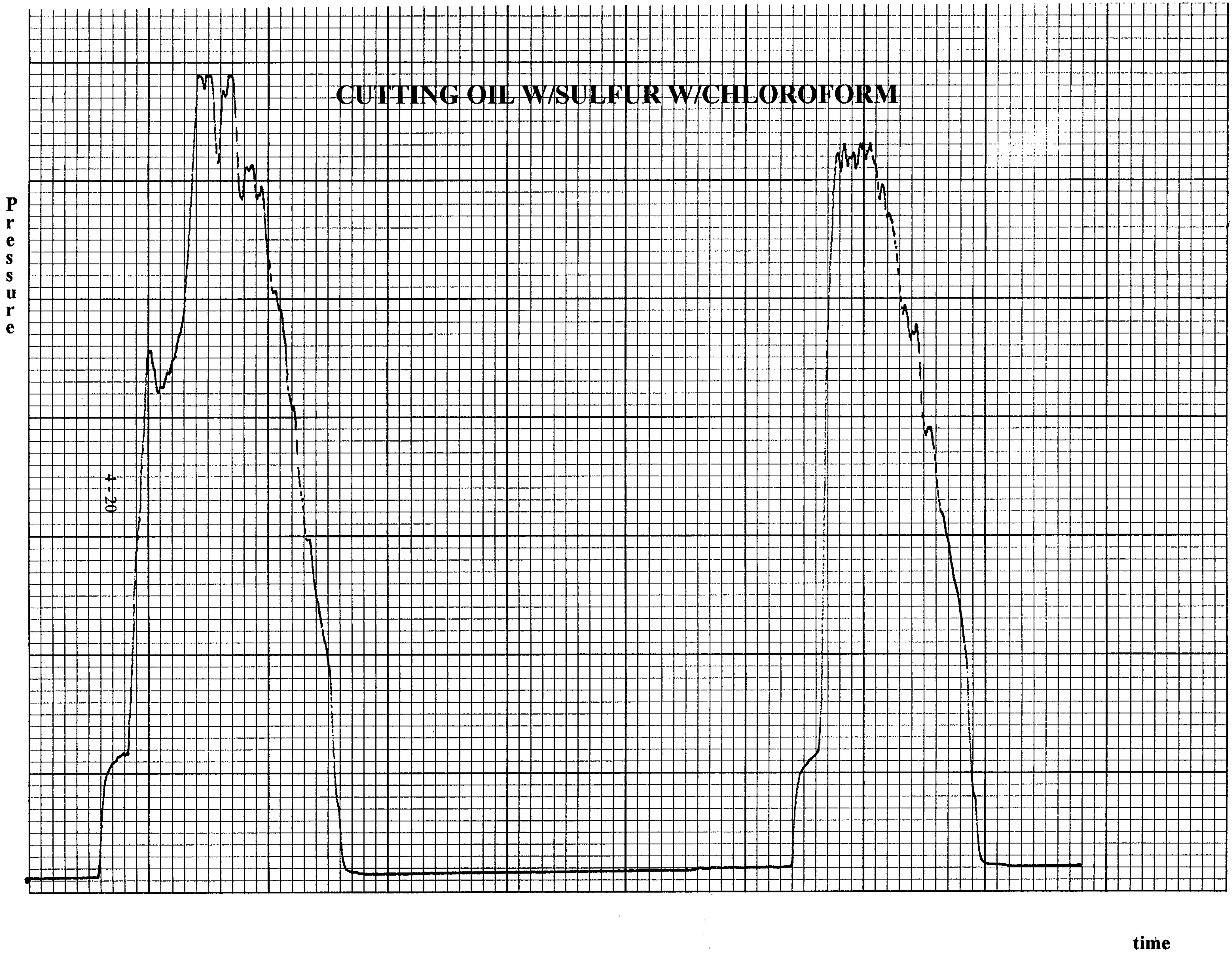
HYDROCARBON FRACTION W/ACETONITRITE

HYDROCARBON FRACTION W/PROPARGYL CHLORIDE

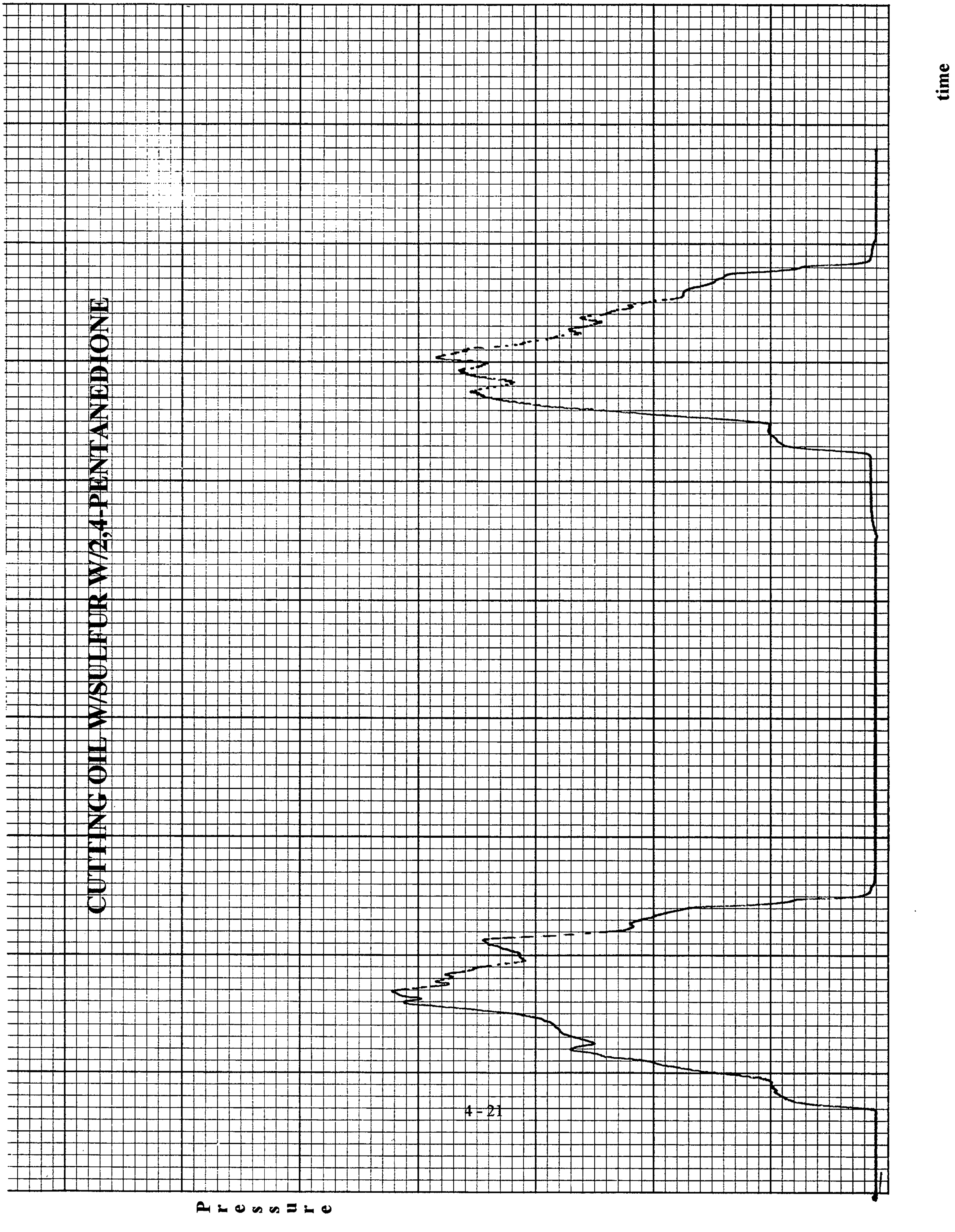


CUTTING OIL W/SULFUR W/1,2-DICHLOROBENZENE

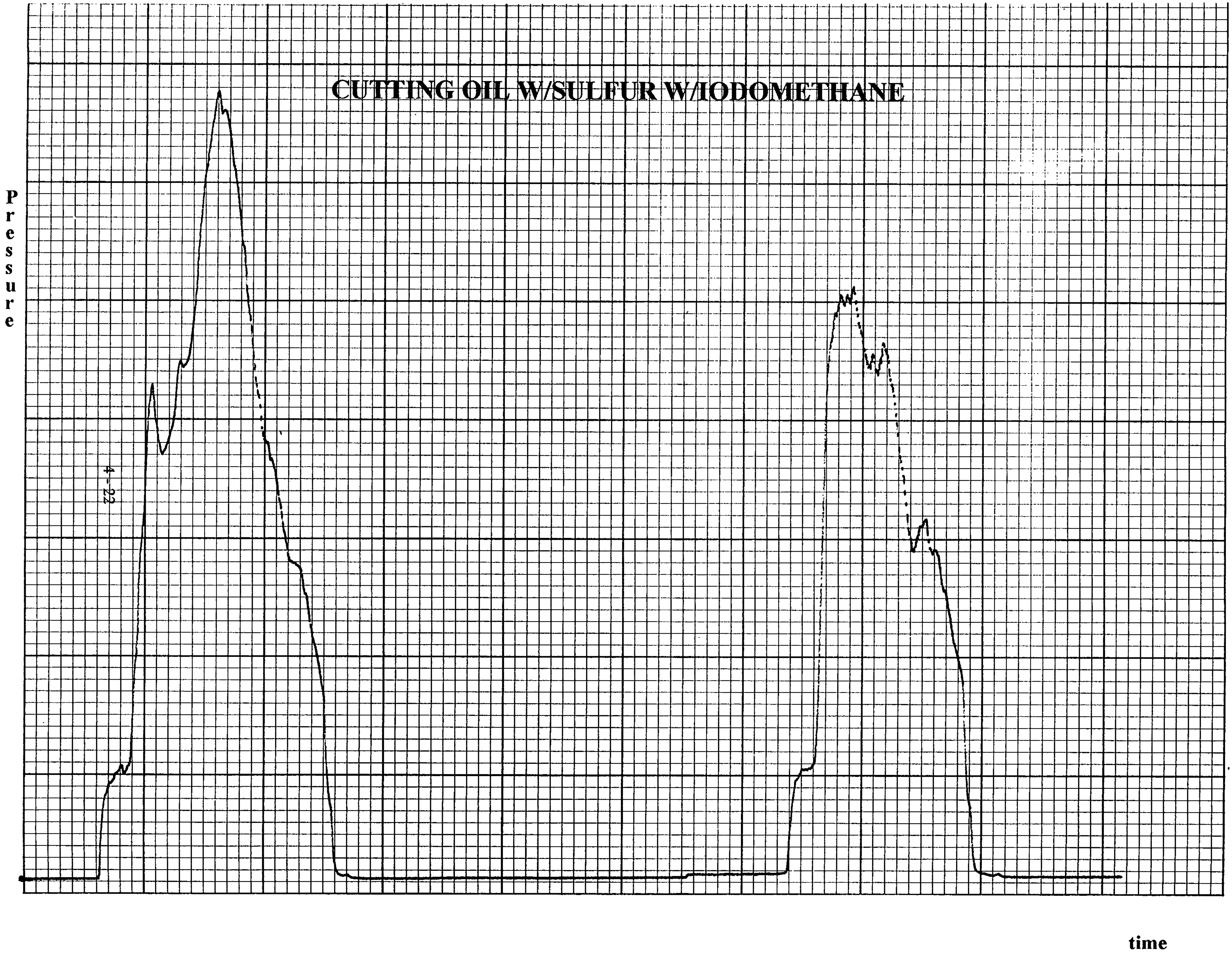




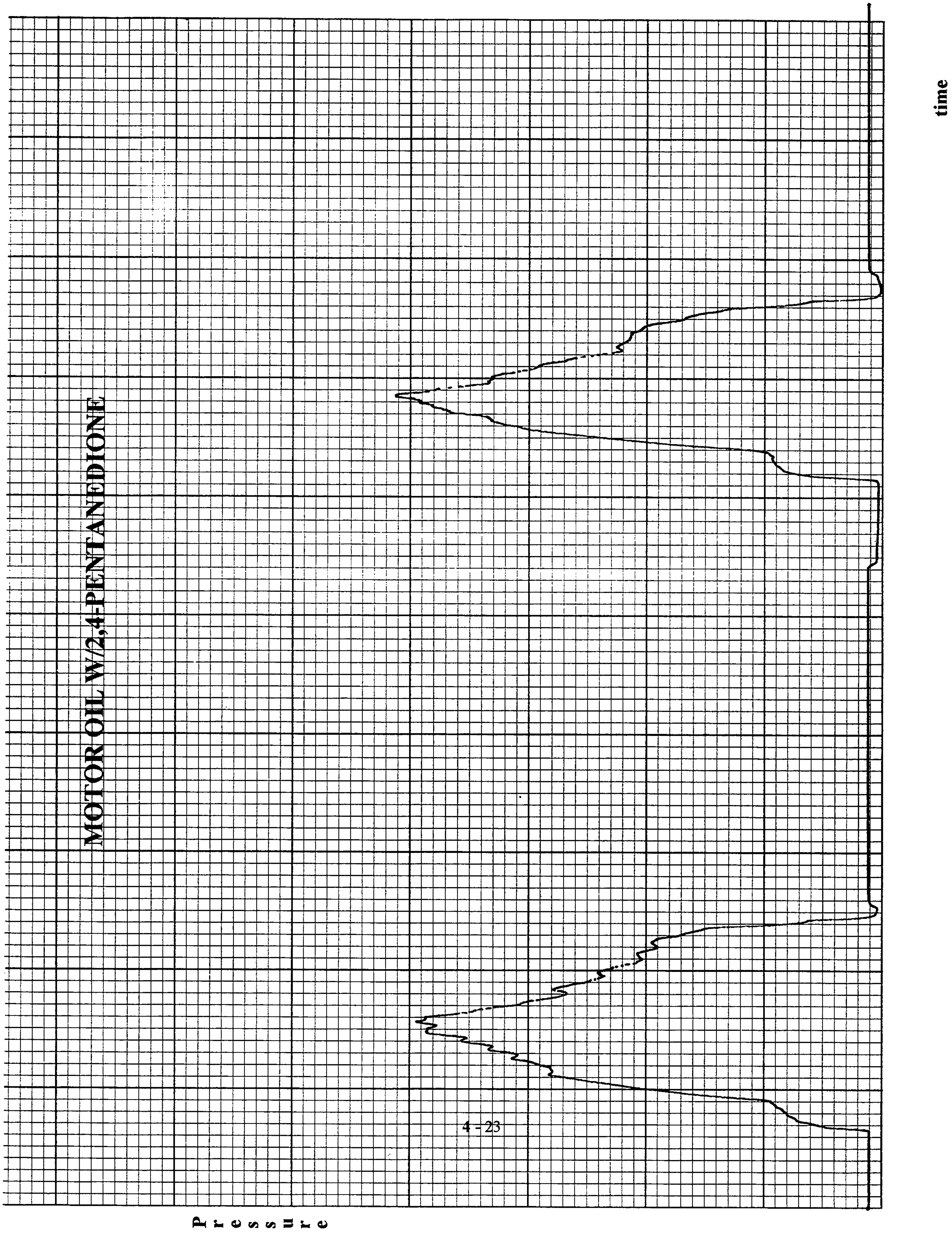
CUTTING OIL W/SULFUR W/CHLOROFORM

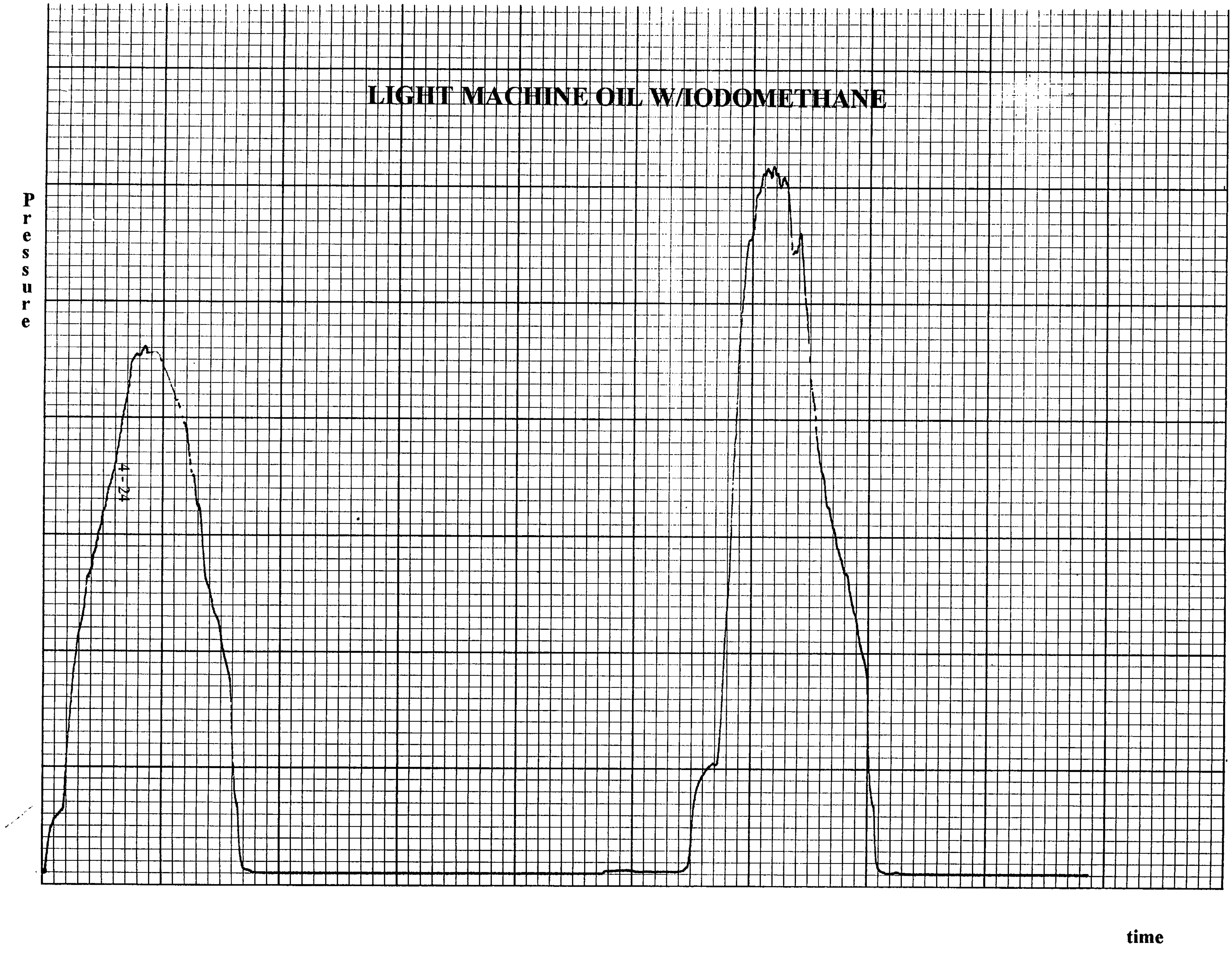


CUTTING OIL W/SULFUR W/2,4-PENTANEDIONE



CUTTING OIL W/SULFUR W/IODOMETHANE


MOTOR OIL W/2,4-PENTANEDIONE
time
Pressure



LIGHT MACHINE OIL W/IODOMETHANE

APPENDIX B.

Contents:

Listing of all doping agents tested.

Listing of doping agents yielding positive results.

The Raman spectra scans from 1100 - 1700 cm^{-1} for lubricated drill bits that showed “hard carbon” patterns.

The Raman spectra scans from 840 - 2000 cm^{-1} for lubricated drill bits that showed “hard carbon” patterns.

APPENDIX B
All materials tested

SAMPLE		BASE	DOPING AGENT	SHOWED
				NOTHING
ANMS	A001	HYDROCARBON FRACTION	1,4-DIOXANE	*
ANMS	A002	HYDROCARBON FRACTION	1,4-DIOXANE	*
ANMS	A101	HYDROCARBON FRACTION	STYRENE	*
ANMS	A201	HYDROCARBON FRACTION	IODOMETHANE	*
ANMS	A301	HYDROCARBON FRACTION	2,4-PENTANEDIONE	*
ANMS	A401	HYDROCARBON FRACTION	PROPARGYL CHLORIDE	*
ANMS	A402	HYDROCARBON FRACTION	PROPARGYL CHLORIDE	
ANMS	A501	HYDROCARBON FRACTION	3-METHYL-2-BUTANONE	*
ANMS	A601	HYDROCARBON FRACTION	1,2-DICHLOROBENZENE	*
ANMS	A701	HYDROCARBON FRACTION	SULFOLANE	*
ANMS	A801	HYDROCARBON FRACTION	ACETONITRITE	
ANMS	A901	HYDROCARBON FRACTION	TETRAMETHYLETHYLENEDIAMINE	*
ANMS	AA01	HYDROCARBON FRACTION	DIMETHYL ACETAMIDE	*
ANMS	AA02	HYDROCARBON FRACTION	DIMETHYL ACETAMIDE	
ANMS	AB01	HYDROCARBON FRACTION	1,1,1-TRICHLOROETHANE	*
ANMS	AC01	HYDROCARBON FRACTION	1,2-DIBROMOETHANE	*
ANMS	AD01	HYDROCARBON FRACTION	CHLOROFORM	*
ANMS	AD02	HYDROCARBON FRACTION	CHLOROFORM	
ANMS	AD03	HYDROCARBON FRACTION	CHLOROFORM	*
ANMS	AE01	HYDROCARBON FRACTION	PHENYL ACETYLENE	
ANMS	B101	LIGHT MACHINE OIL	STYRENE	*
ANMS	B201	LIGHT MACHINE OIL	IODOMETHANE	
ANMS	B301	LIGHT MACHINE OIL	2,4-PENTANEDIONE	
ANMS	B601	LIGHT MACHINE OIL	1,2-DICHLOROBENZENE	*
ANMS	B701	LIGHT MACHINE OIL	SULFOLANE	*
ANMS	B801	LIGHT MACHINE OIL	ACETONITRITE	*
ANMS	BA01	LIGHT MACHINE OIL	DIMETHYL ACETAMIDE	*
ANMS	BD01	LIGHT MACHINE OIL	CHLOROFORM	*
ANMS	C101	30 W MOTOR OIL	STYRENE	*
ANMS	C401	30 W MOTOR OIL	PROPARGYL CHLORIDE	*
ANMS	C601	30 W MOTOR OIL	1,2-DICHLOROBENZENE	*
ANMS	CA01	30 W MOTOR OIL	DIMETHYL ACETAMIDE	*
ANMS	CD01	30 W MOTOR OIL	CHLOROFORM	*
ANMS	D001	CUTTING OIL W/SULFUR	1,4-DIOXANE	*
ANMS	D002	CUTTING OIL W/SULFUR	1,4-DIOXANE	*
ANMS	D003	CUTTING OIL W/SULFUR	1,4-DIOXANE	*
ANMS	D101	CUTTING OIL W/SULFUR	STYRENE	*
ANMS	D201	CUTTING OIL W/SULFUR	IODOMETHANE	*
ANMS	D203	CUTTING OIL W/SULFUR	IODOMETHANE	*
ANMS	D301	CUTTING OIL W/SULFUR	2,4-PENTANEDIONE	*

APPENDIX B

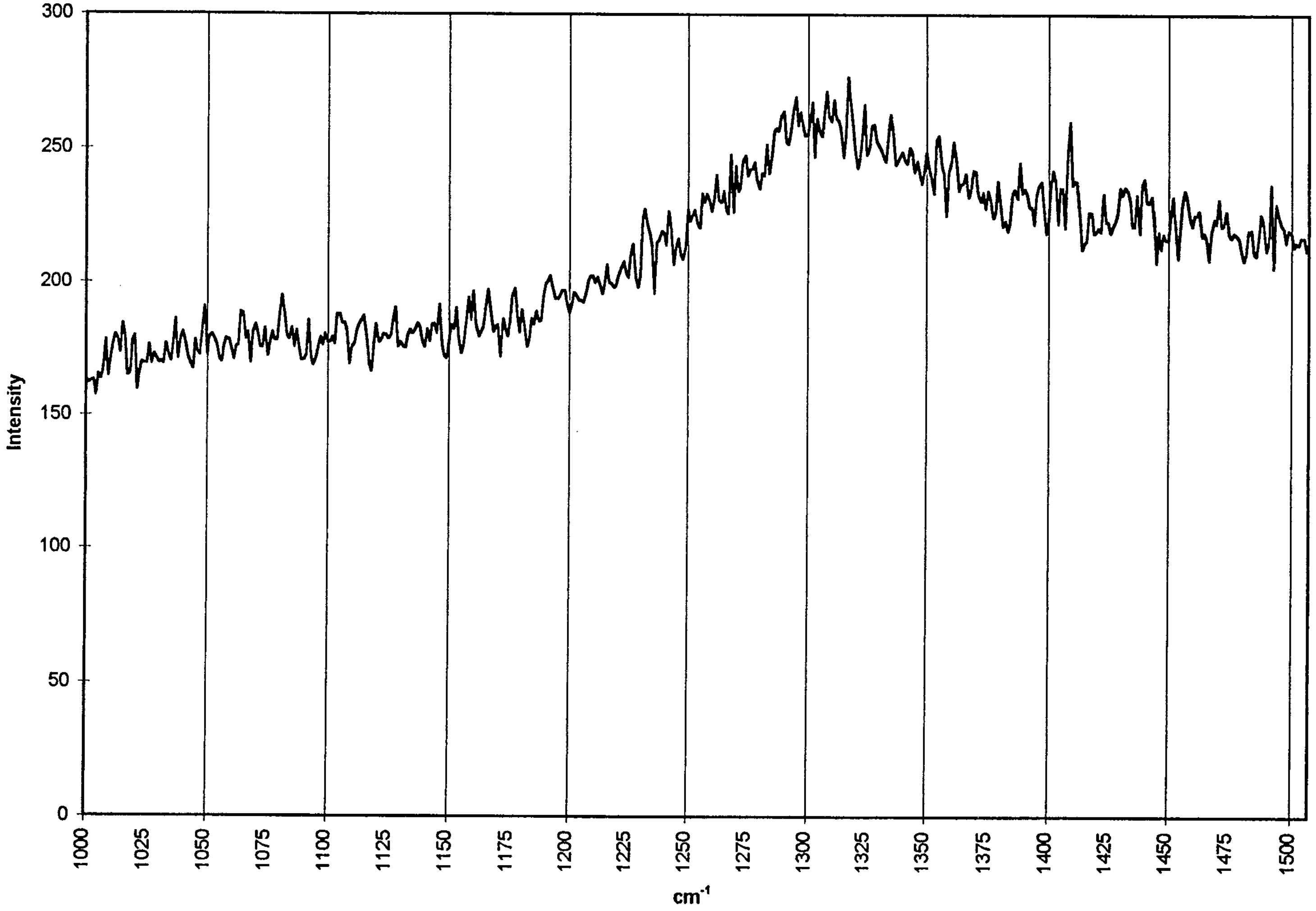
All materials tested

ANMS	D302	CUTTING OIL W/SULFUR	2,4-PENTANEDIONE	*
ANMS	D401	CUTTING OIL W/SULFUR	PROPARGYL CHLORIDE	*
ANMS	D501	CUTTING OIL W/SULFUR	3-METHYL-2-BUTANONE	*
ANMS	D601	CUTTING OIL W/SULFUR	1,2-DICHLOROBENZENE	
ANMS	D603	CUTTING OIL W/SULFUR	1,2-DICHLOROBENZENE	
ANMS	D701	CUTTING OIL W/SULFUR	SULFOLANE	*
ANMS	D703	CUTTING OIL W/SULFUR	SULFOLANE	*
ANMS	D801	CUTTING OIL W/SULFUR	ACETONITRITE	*
ANMS	D901	CUTTING OIL W/SULFUR	TETRAMETHYLETHYLENEDIAMINE	*
ANMS	DA01	CUTTING OIL W/SULFUR	DIMETHYL ACETAMIDE	*
ANMS	DAT2	CUTTING OIL W/SULFUR	DIMETHYL ACETAMIDE	*
ANMS	DAT3	CUTTING OIL W/SULFUR	DIMETHYL ACETAMIDE	*
ANMS	DATU	CUTTING OIL W/SULFUR	DIMETHYL ACETAMIDE	*
ANMS	DBO1	CUTTING OIL W/SULFUR	1,1,1-TRICHLOROETHANE	*
ANMS	DBPU	CUTTING OIL W/SULFUR	1,1,1-TRICHLOROETHANE	*
ANMS	DC01	CUTTING OIL W/SULFUR	1,2-DIBROMOETHANE	*
ANMS	DC02	CUTTING OIL W/SULFUR	1,2-DIBROMOETHANE	*
ANMS	DD01	CUTTING OIL W/SULFUR	CHLOROFORM	
ANMS	DD02	CUTTING OIL W/SULFUR	CHLOROFORM	*
ANMS	DD03	CUTTING OIL W/SULFUR	CHLOROFORM	*
ANMS	DE01	CUTTING OIL W/SULFUR	PHENYL ACETYLENE	*
XNMS	AR01	CALIBRATION		

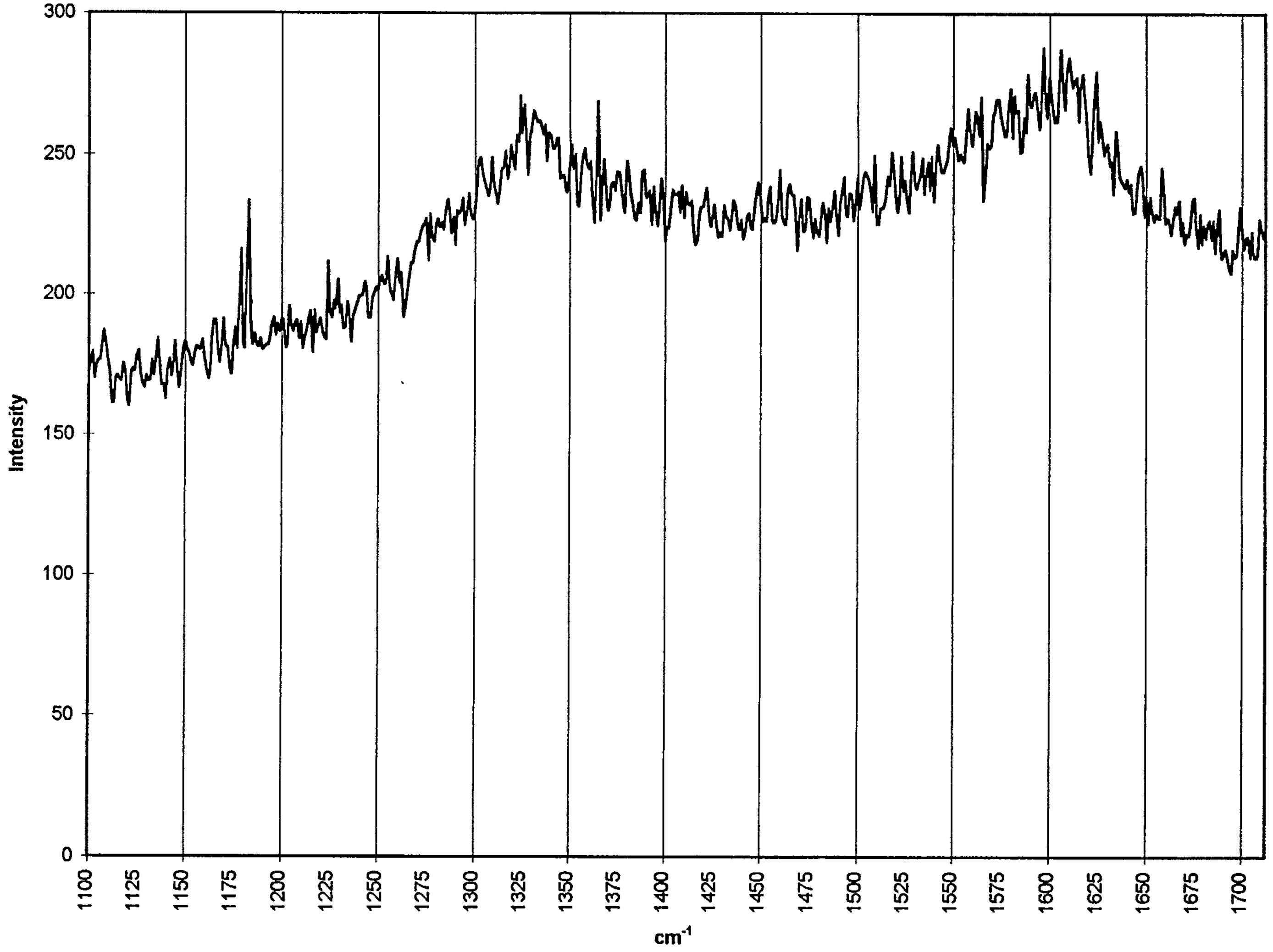
Positive Results Listing

SAMPLE		BASE	DOPING AGENT
ANMS	A402	HYDROCARBON FRACTION	PROPARGYL CHLORIDE
ANMS	A801	HYDROCARBON FRACTION	ACETONITRITE
ANMS	AA02	HYDROCARBON FRACTION	DIMETHYL ACETAMIDE
ANMS	AD02	HYDROCARBON FRACTION	CHLOROFORM
ANMS	AE01	HYDROCARBON FRACTION	PHENYL ACETYLENE
ANMS	B201	LIGHT MACHINE OIL	IODOMETHANE
ANMS	B301	LIGHT MACHINE OIL	2,4-PENTANEDIONE
ANMS	D601	CUTTING OIL W/SULFUR	1,2-DICHLOROBENZENE
ANMS	D603	CUTTING OIL W/SULFUR	1,2-DICHLOROBENZENE
ANMS	DD01	CUTTING OIL W/SULFUR	CHLOROFORM

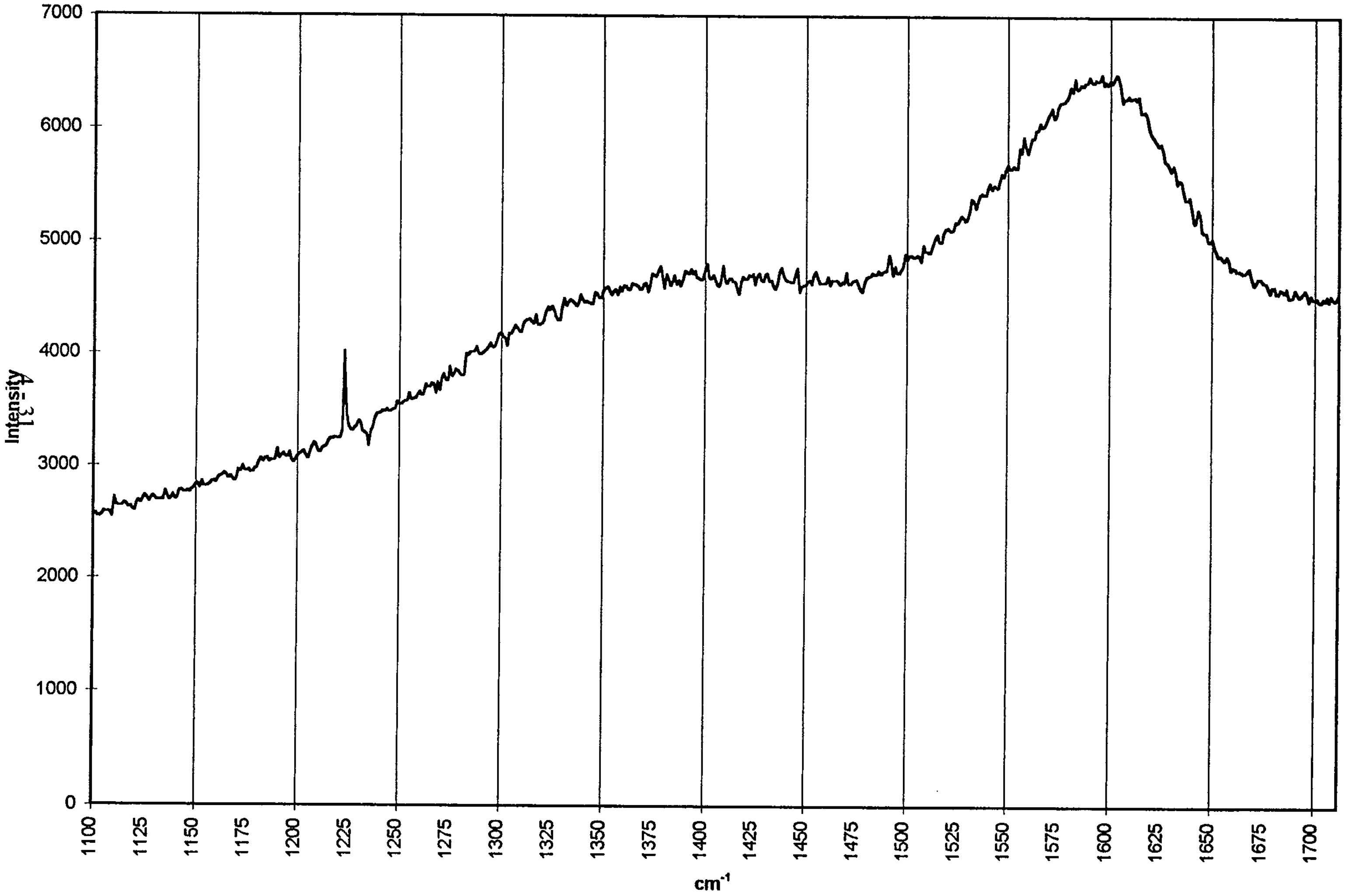
HYDROCARBON FRACTION W/PROPAGYL CHLORIDE



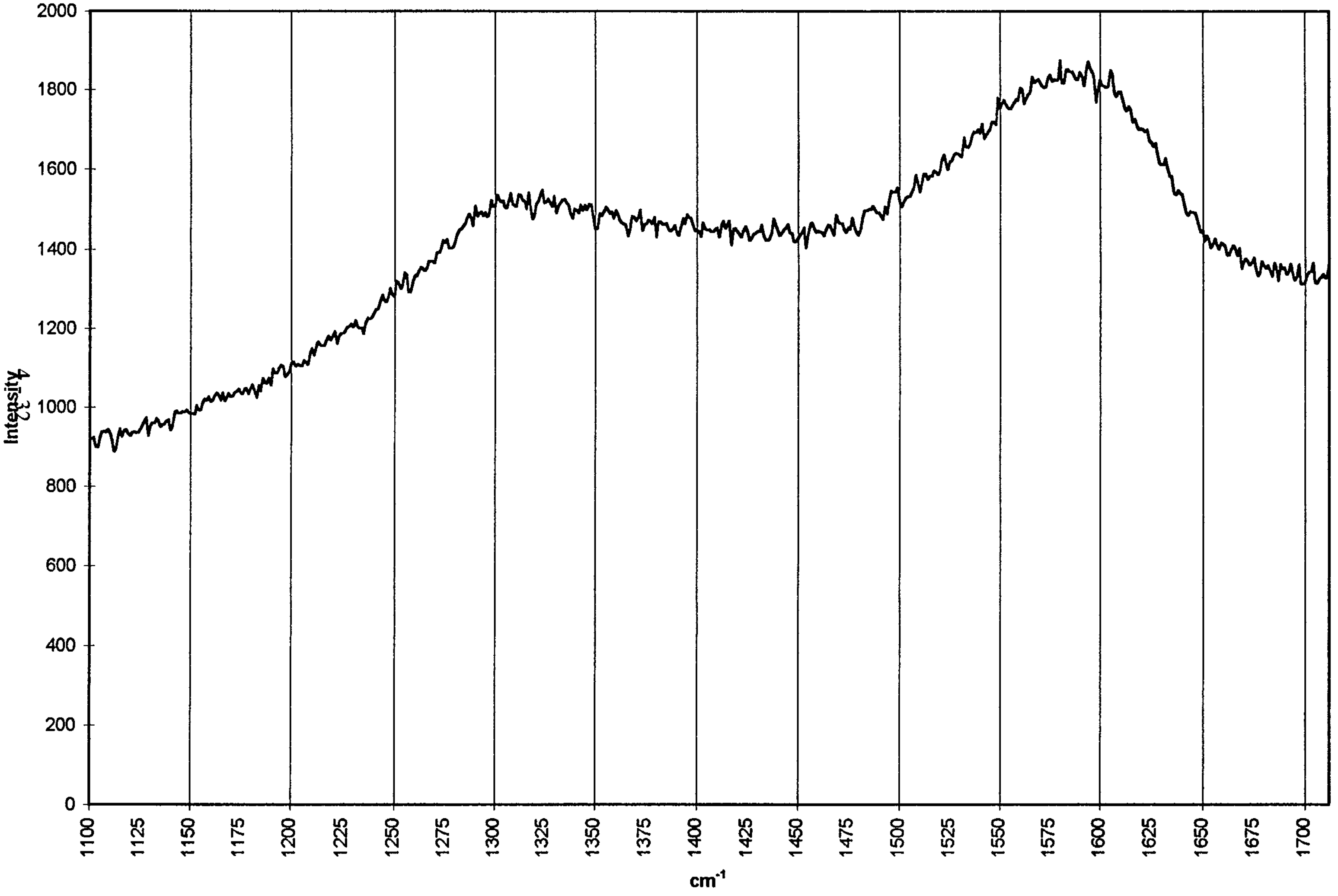
HYDROCARBON FRACTION W/CHLOROFORM



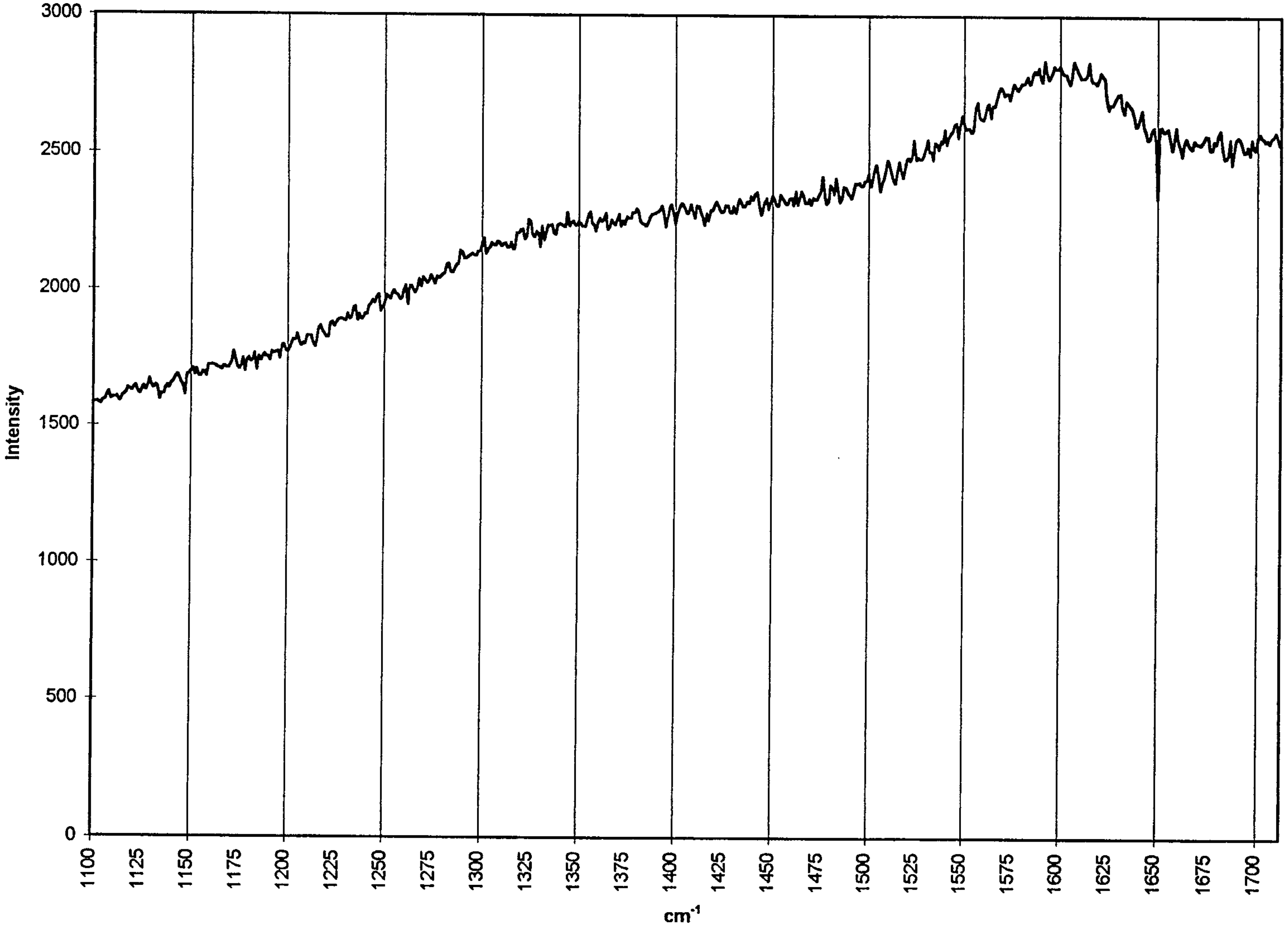
HYDROCARBON FRACTION W/DIMETHYL ACETAMIDE



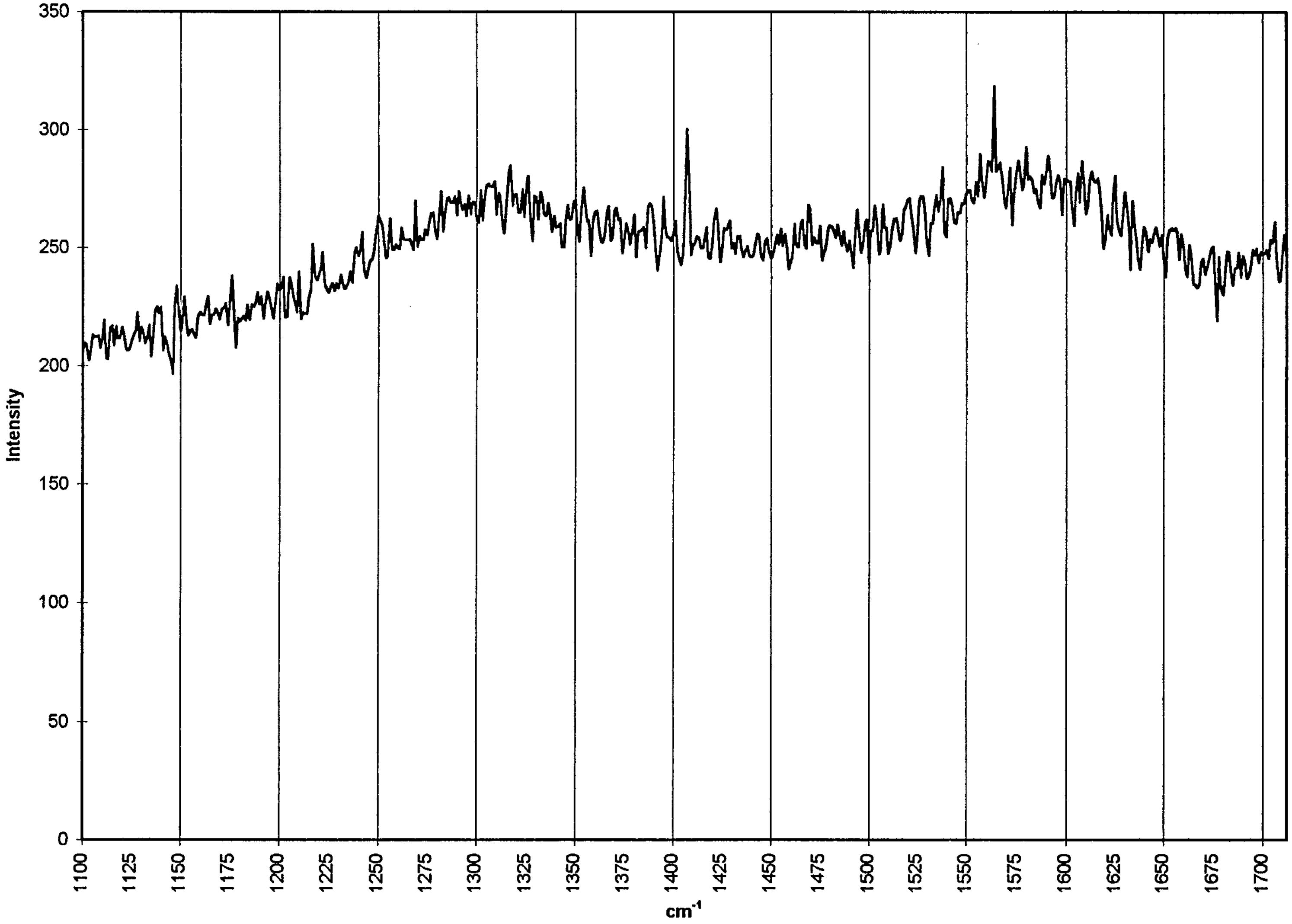
GASOLINE FRACTION W/ACETONITRITE



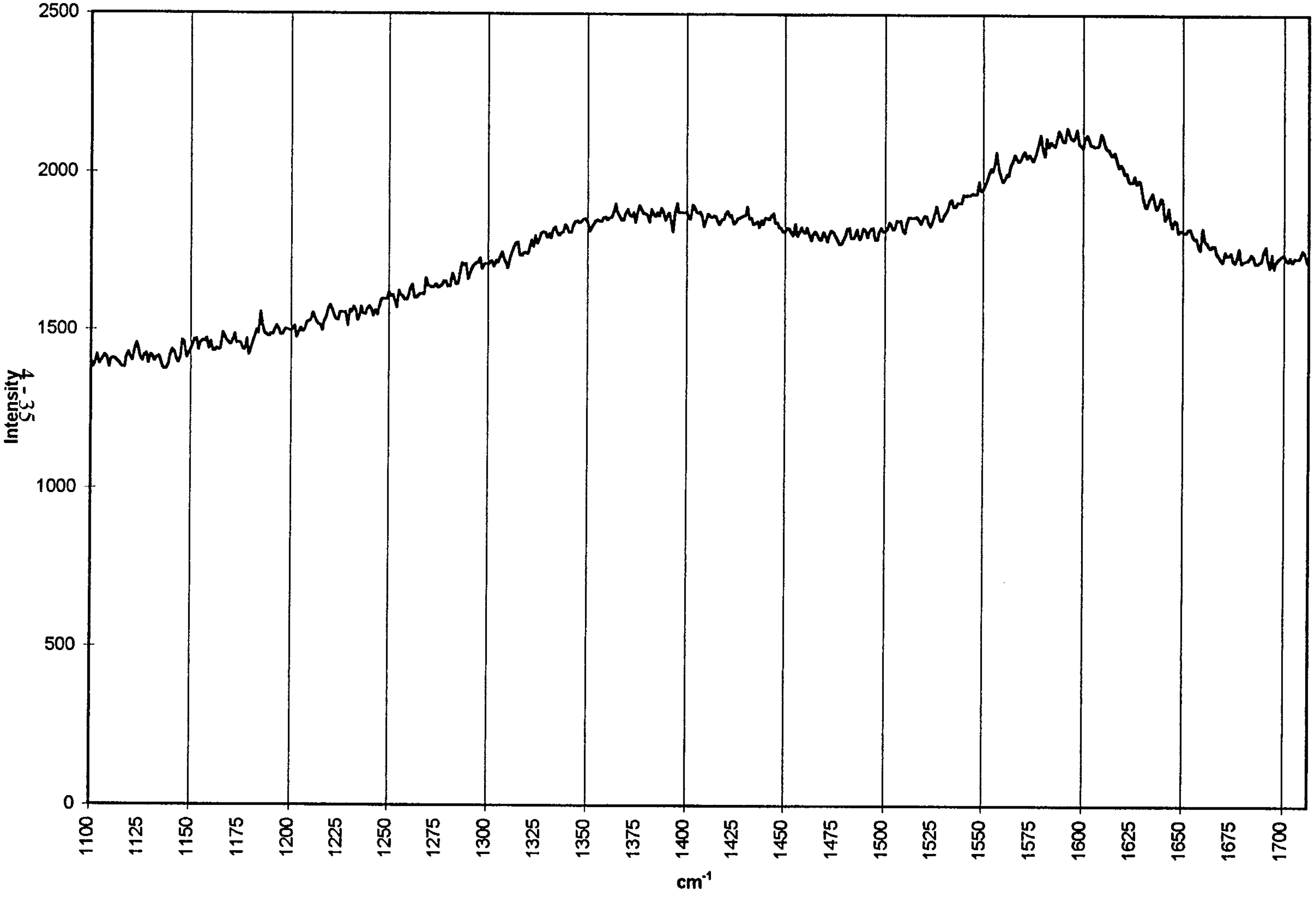
LIGHT MACHINE OIL
W/2,4-PENTANEDIONE



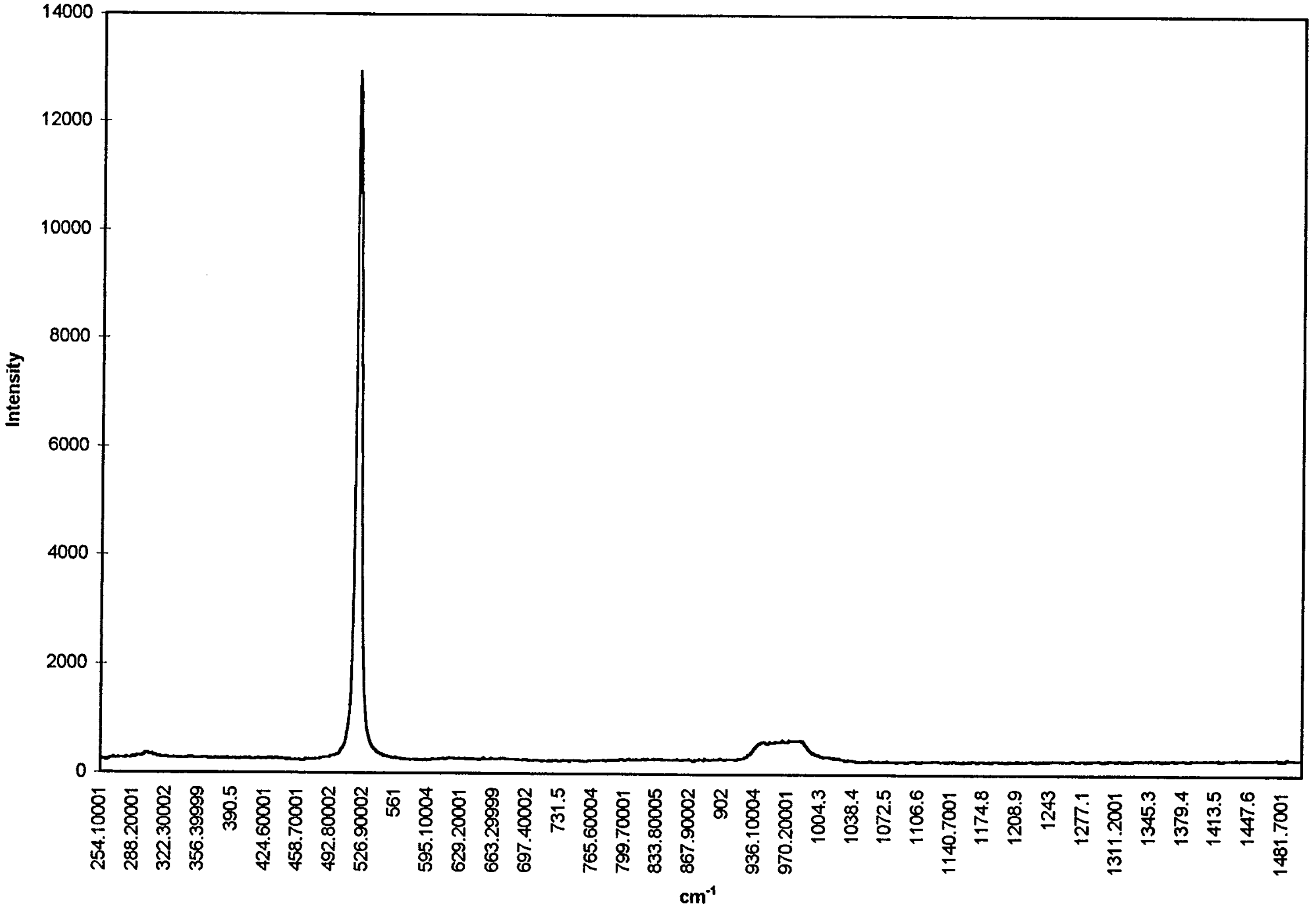
LIGHT MACHINE OIL W/SODOMETHANE



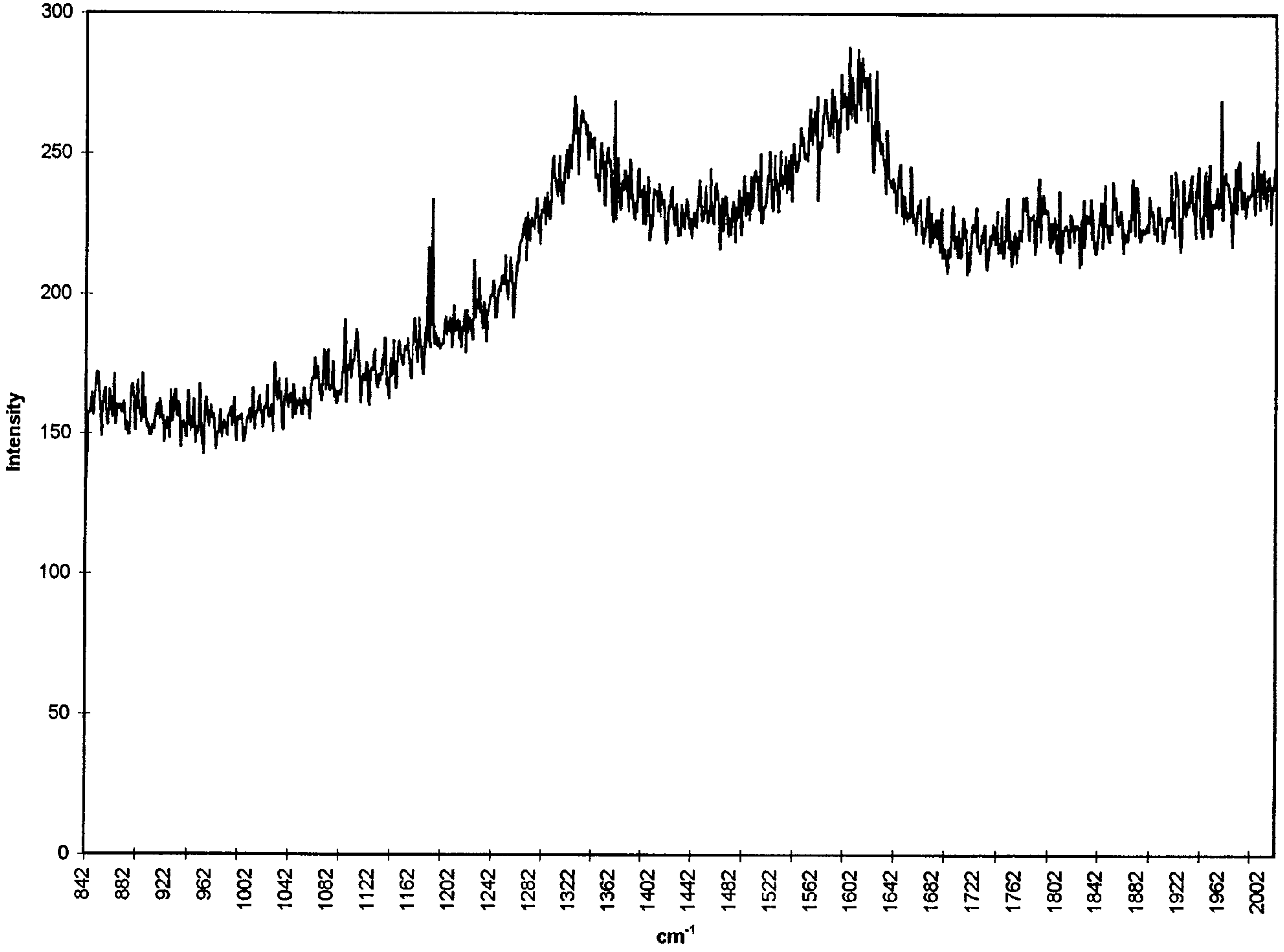
CUTTING OIL W/SULFER
W/1,2-DICHLOROBENZENE



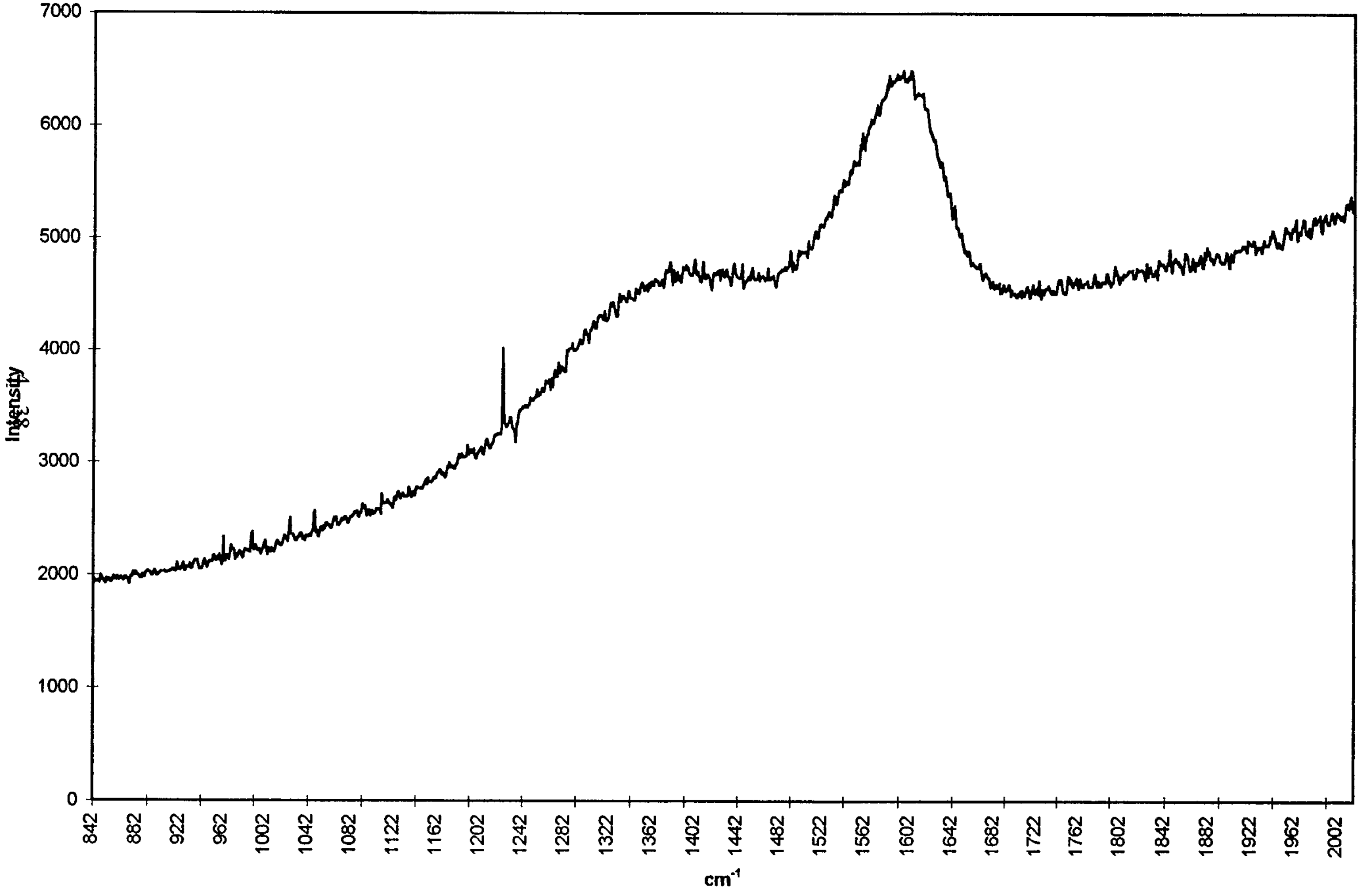
XMNSAR01
CALIBRATION



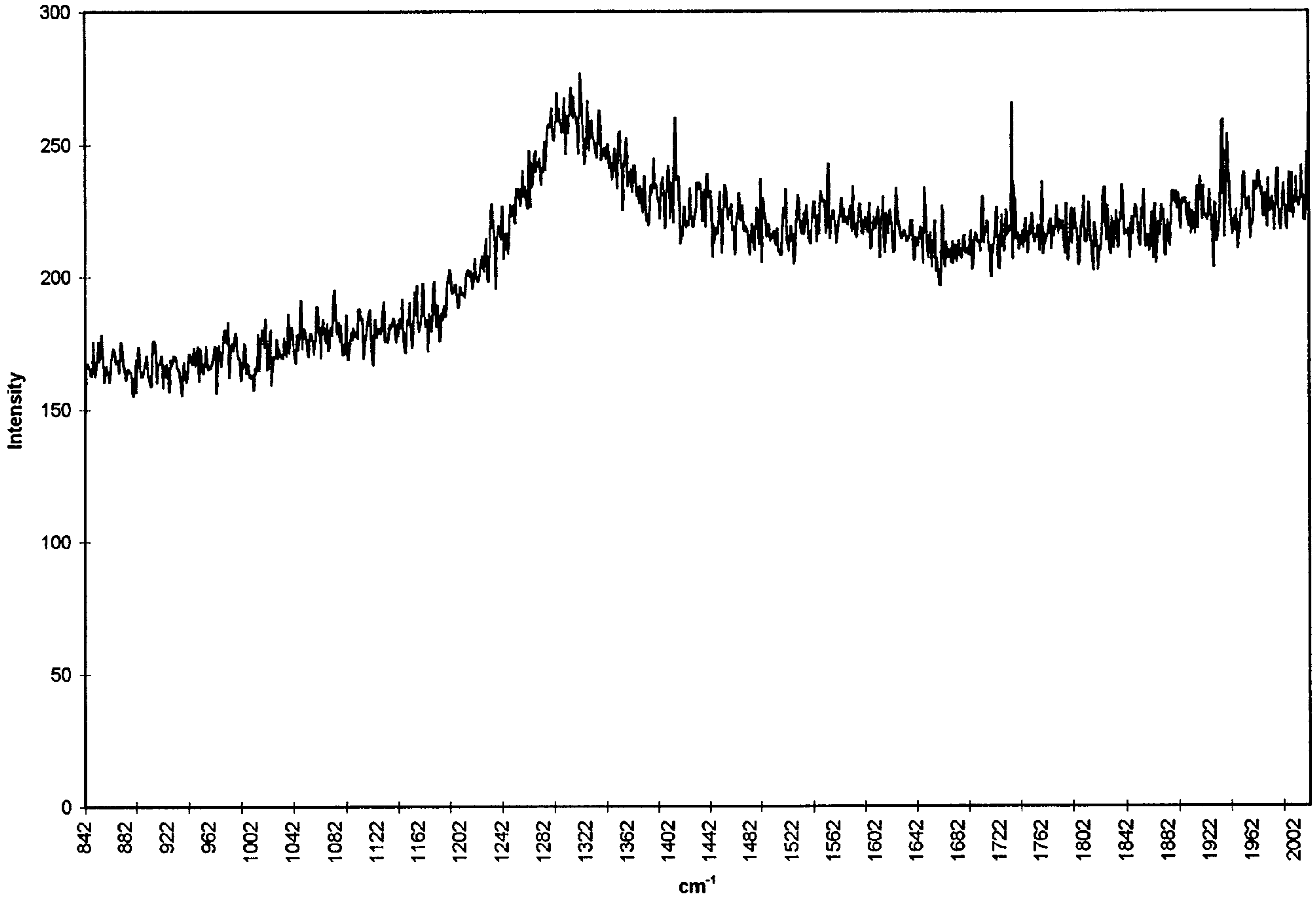
HYDROCARBON FRACTION W/CHLOROFORM



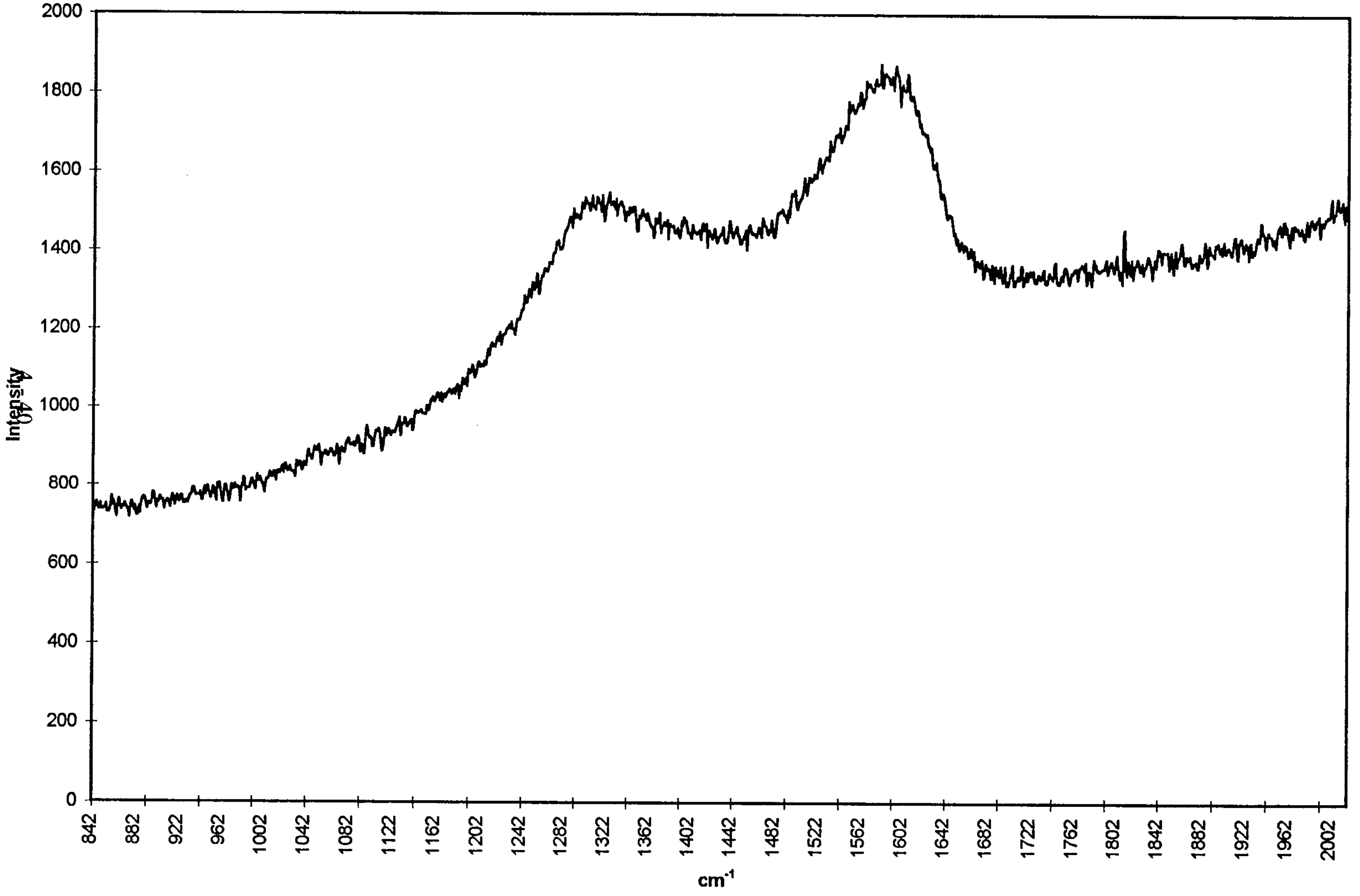
HYDROCARBON FRACTION
W/DIMETHYL ACETAMIDE



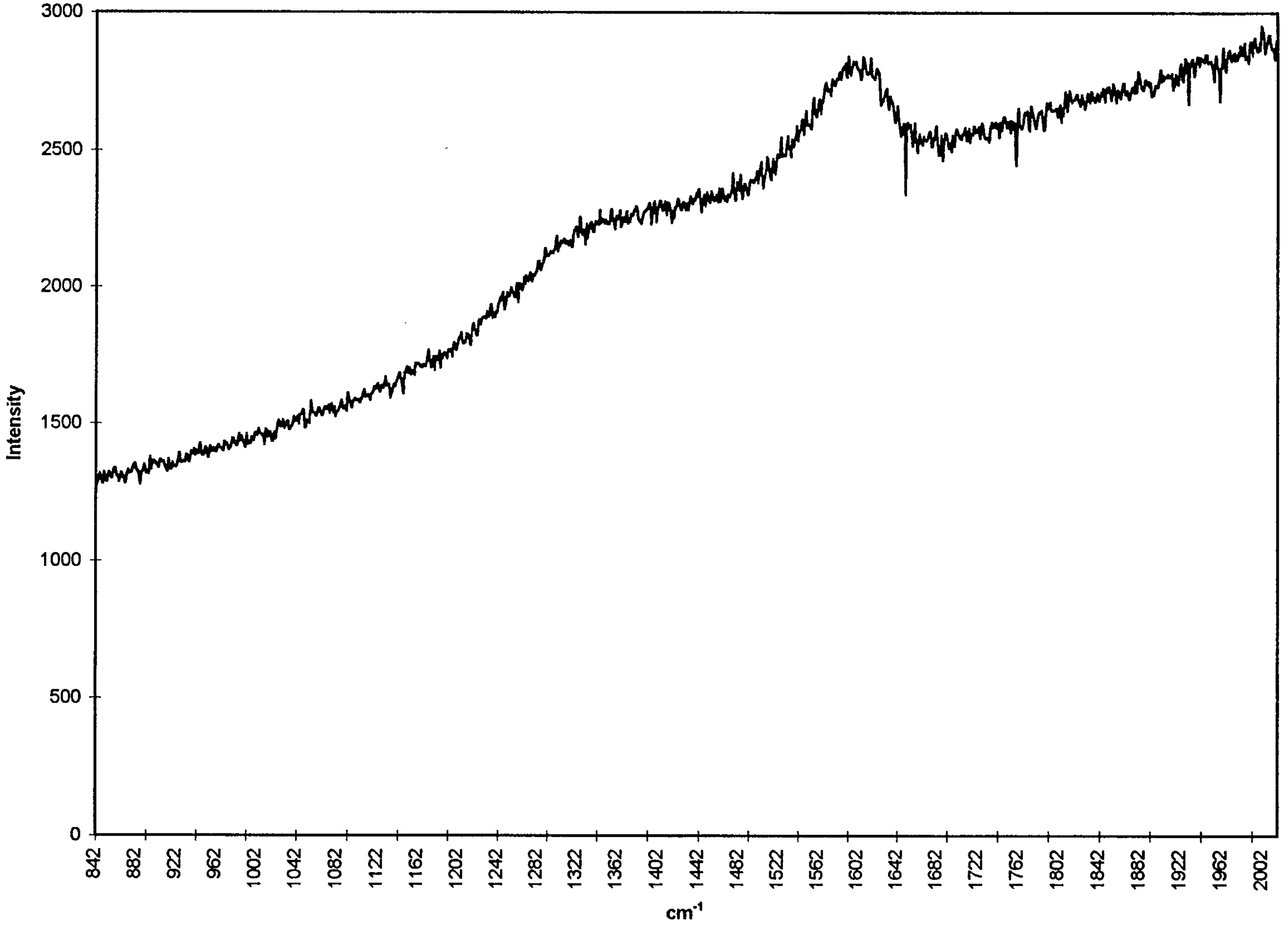
HYDROCARBON FRACTION
W/PROPAGYL CHLORIDE



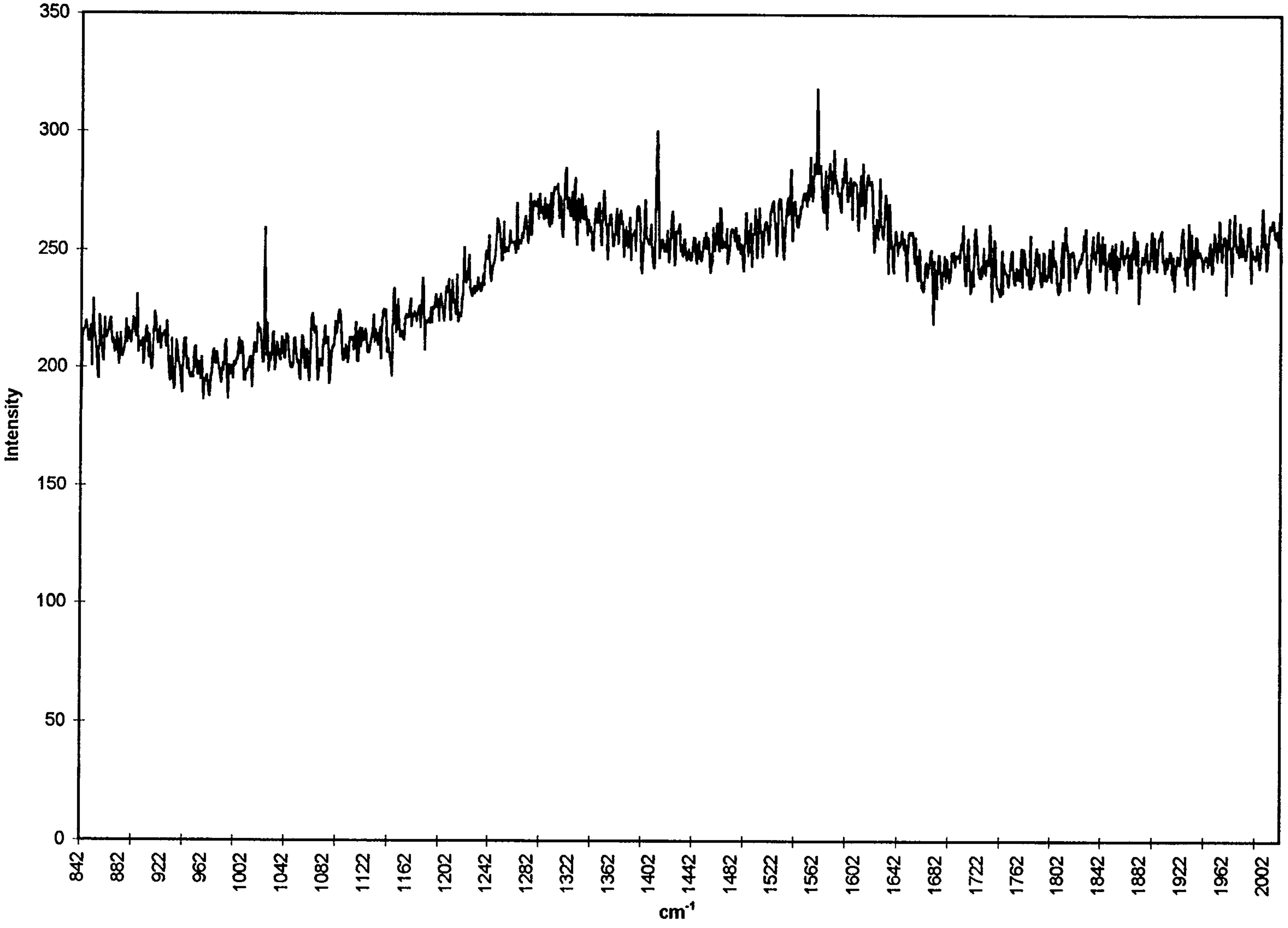
HYDROCARBON FRACTION W/ACETONITRITE



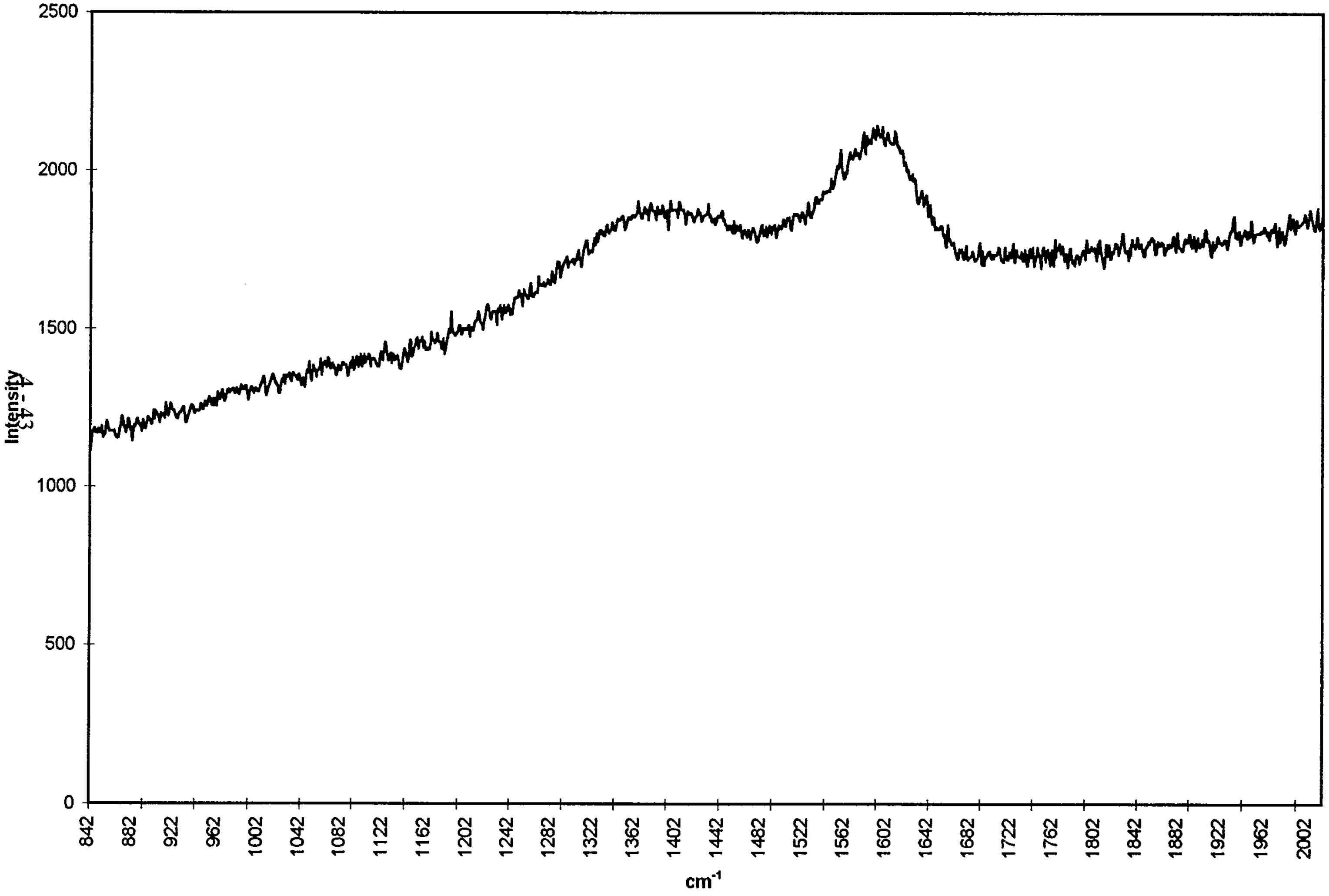
LIGHT MACHINE OIL W/2,4-PENTANEDIONE



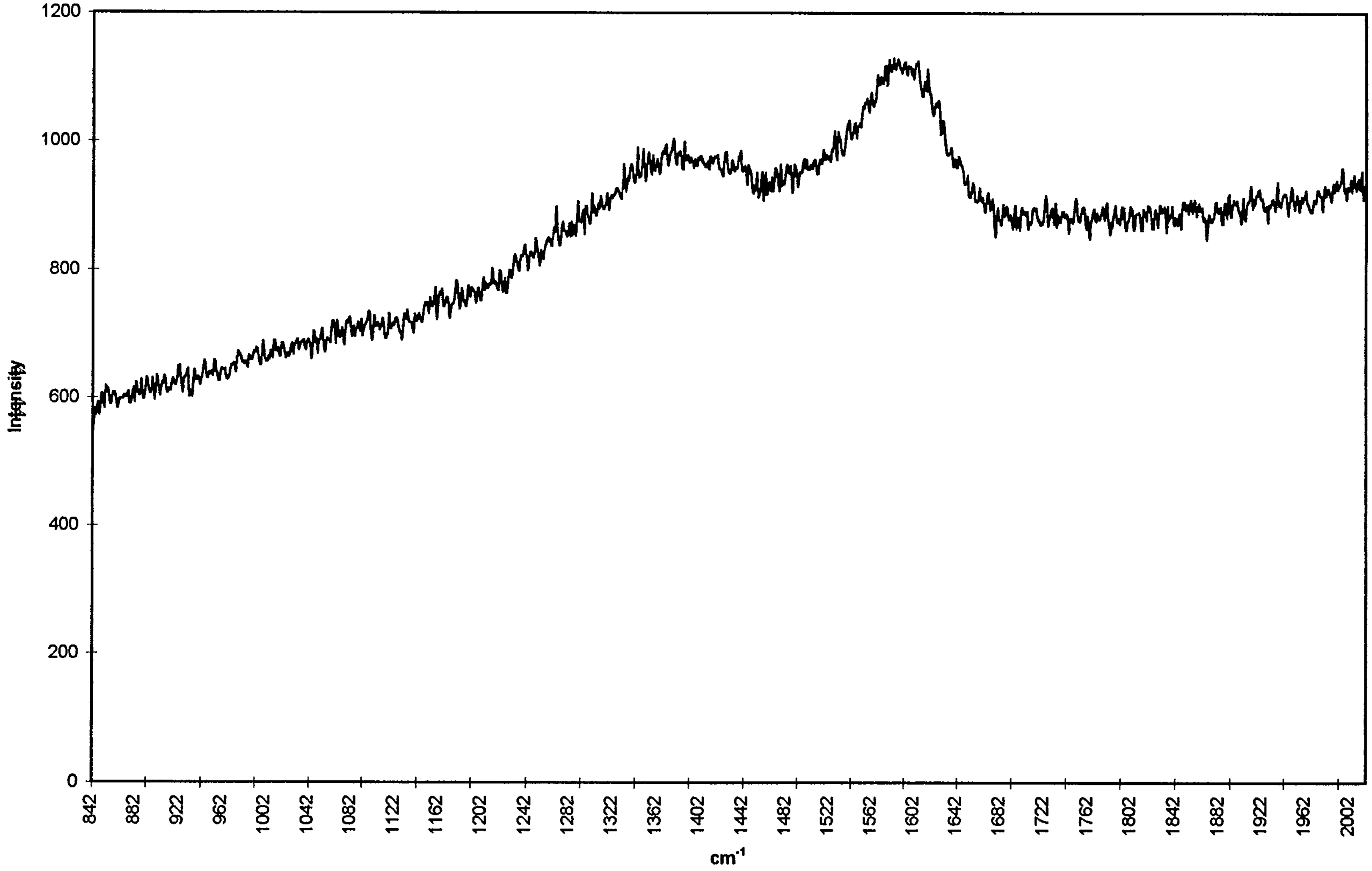
LIGHT MACHINE OIL W/SODOMETHANE



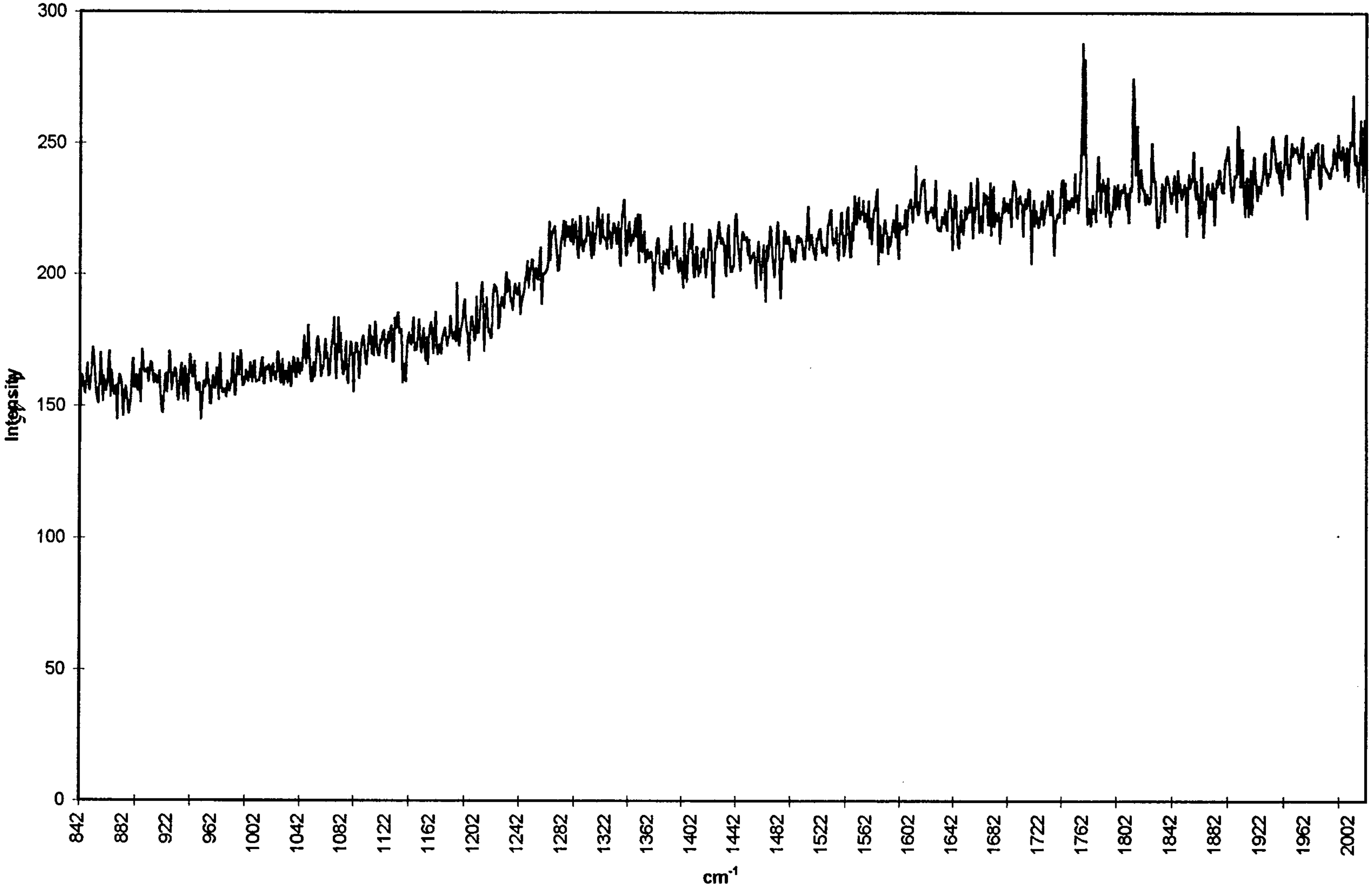
CUTTING OIL W/SULFER
W/1,2-DICHLOROBENZENE



CUTTING OIL W/SULFER
W/1,2-DICHLOROBENZENE



CUTTING OIL W/SULFER W/CHLOROFORM



ARO130

A Performance Study of Plasma Source Ion-Implanted Tools Versus High-Speed Steel Tools

Brian K. Lambert and Roger H. Woods

New Mexico State University

Las Cruces, NM 88003

Abstract

The performance of plasma source ion-implanted high-speed steel tools were compared with untreated high-speed steel tools in a test modeled after actual manufacturing usage. The variables of interest were tool wear and the resulting surface finish of the work piece. A set of 12 M2 high-speed steel turning tools were implanted with N^+, resulting in a retained dosage of 2.5×10^{17} N-at/cm^2 using plasma source ion implantation. The 12 tools where compared to 12 unimplanted tools using a simple turning operation on 4140 steel. The implanted tools exhibited an increase in wear resistance by a factor of 1.5 over the unimplanted tools, and implantation exhibited no effect on the resulting surface finish.

I. Introduction

Tool wear and tool failure are areas researched in great detail to improve the manufacturing process. New coatings and materials have been developed to lengthen the life of metal cutting tools. Unfortunately, these processes can produce harmful by-products and adversely affect the environment. Ion implantation has been shown to improve the wear characteristics of metals without producing environmentally harmful by-products.

Throughout the machining process, the dimensions of a tool change because of wear to its contact surfaces. Manufacturing engineers want to predict that wear to hold tolerances and surface finishes throughout the process. Being able to predict the wear of the tool also allows the engineer to determine the life of the tool, thereby avoiding a catastrophic tool failure and possible damage to the part, the operator, and machinery. The engineer can effectively model this wear by understanding the variables that determine tool life and wear rate: tool material, tool geometry, machining variables, and work piece variables.

The tool material used is determined by the material to be machined and the process under which the material goes. The development of new, stronger, harder, tougher tool materials has been an area of great importance in manufacturing. The tool material must have certain traits and characteristics to resist wear and tool failure.

Typically, an increase in the hardness of the tool material decreases its fracture toughness. However, the hardness increases, the material may be too hard to properly form into the necessary tool shape. Coatings have been developed to increase wear resistance but may not remain adhered to the surface of the tool and can produce harmful by-products. The engineer continuously balances the speeds, feeds, and tool material to achieve the lowest cost possible. Speeds and feeds must be kept low enough to provide for a minimum acceptable tool life; otherwise, the cost of changing the tool may outweigh the productivity gains in increased cutting speed. The advent of ion implantation technology may have solved some of these materials problems.

A. Tool Wear

Abrasive wear is the primary wear mechanism during steady state wear at low temperatures and is measured by the flank wear of the tool. Plastic deformation occurs at high temperature and high cutting pressures. The tool deforms under the pressure and temperature, rounding the cutting edge of the tool. Chemical decomposition weakens the bonds between minute tool segments, resulting in the particles being pulled off the tool. As the work-hardened chip passes by the tool, the asperites in the tool surface may be welded to the chip and effectively pulled off. This wear occurs on the top of the tool, resulting in crater wear.

Tool wear is measured by the area of flank wear or the depth of crater wear on a tool. Flank wear occurs on the front surface or flank where the tool contacts the work piece material. Crater wear occurs on the top of the tool where the work hardened chip contacts the tool surface. Tool failure occurs when the tool can no longer produce parts to specifications because of the wear to the tool.[1]

Tool life is determined by a set amount of flank wear that occurs before the tool is unable to hold tolerance or could experience catastrophic tool failure. The generally accepted value for maximum wear land is 0.025 to 0.030 inch.[1,2] Wear land is determined by measuring the height of tool wear (W_f or F_w) along the contact length (l or x) as illustrated in Fig. 1 from DeGarmo[2].

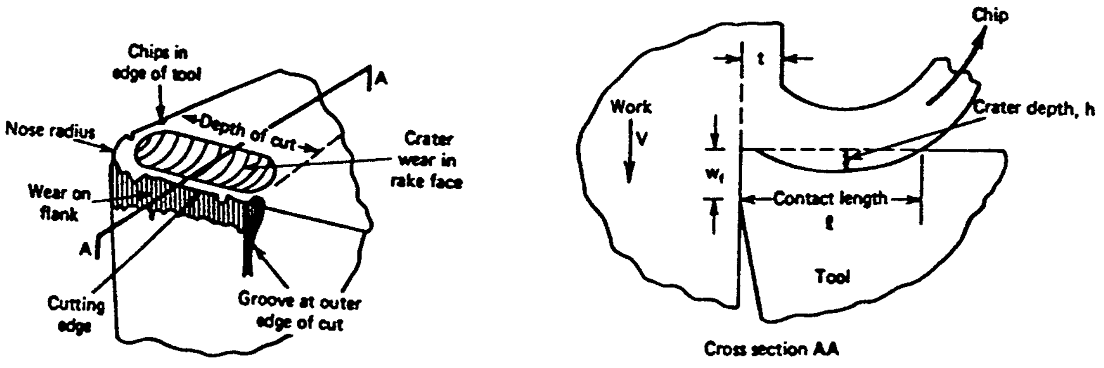


Figure 1. Tool wear (DeGarmo[2])

B. Surface Finish

Attaining the proper surface quality is an important part of producing quality parts. The close tolerances required in today's complex systems require that the finishes be achieved according to the specifications each and every time. Surface roughness changes as the machining process progresses because of the wear that occurs on the flank of the tool, changing its dimensions. This tool wear can progress to the point where the tool is not producing parts with the necessary finish or to the necessary tolerances.

C. Ion Implantation

Ion implantation modifies surface sensitive properties of different materials. Ion implantation increases the wear resistance, corrosion resistance, fatigue resistance and hardness of many materials. Implantation is used both to harden cutting tools and as a surface finish on parts that are subjected to heavy wear or extreme conditions.

Scientists discovered that the ability to introduce any ion into the surface region of a substrate without the constraints of thermodynamic equilibrium could have distinct advantages in metal processing. Using ion implantation, a certain impurity can be introduced into the surface layer in a very pure, clean, controllable, and reproducible way.[3,4] The advent of ion implantation as a surface treatment has now made possible the formation of compounds outside the usual restraints of a metal's thermodynamic equilibrium.

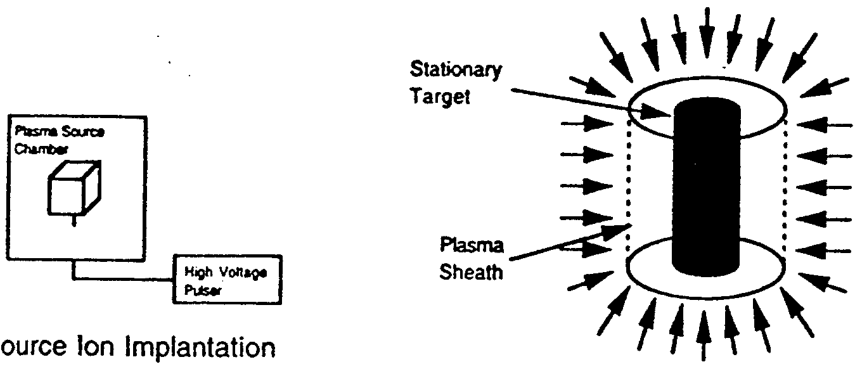
Unlike coating techniques, ion implantation changes the molecular and chemical structure of the surface without adding any volume; therefore, dimensional tolerances are maintained.[5,6] Normally, the real weakness in deposition or coating techniques lies in the adhesion of a binder or glue. The surface layer in ion implantation becomes an integral part of the material without the discontinuities or adhesion problems found in other processes.[4,7]

The temperature of the target can be controlled during implantation and, therefore, introduce impurities during an earlier metal stage in crystal growth. The ability to control the temperature allows the treatment to be done at room temperatures, and the bulk properties of the metal remain unchanged.[8] Parts and tools can be produced using cheaper base metals that are easily formed and then implanted to produce the desired increases in hardness, corrosion resistance, etc.

Until recently, the majority of the research and development outside the semi-conductor industry had been in academia rather than in industry.[9] The emphasis in implantation research has been on optimizing the process and determining the proper implantation voltages, doses, and dopants. The results of these tests have been investigated using pin-on-disk tests and other laboratory-based experiments. The maturation of this technology has now allowed scientists to investigate the practical applications of ion implantation.

Ion implantation is energy deposition in a target material. Ions are produced and then accelerated towards a target material where their interactions produce the resulting chemical and physical property changes. Ions penetrate the work piece because of their high kinetic energy that is gained by accelerating them in an magnetic field.[10] Through elastic collision, the ions displace the target atoms from their lattice positions, creating a distorted lattice structure.[11] The ion's energy carries through the voids in the lattice, creating distortions as it collides with atoms until it comes to rest in a vacancy in the lattice structure.

Plasma source ion implantation was developed to implant irregularly shaped surfaces with an equal dose of ions. The target is placed directly in the plasma source (usually nitrogen) and pulsed to high negative potential relative to the chamber walls. The ions are accelerated normal to the target surface, across the plasma sheath, impacting the target normal to the surface.[7,8,10,12]



Plasma Source Ion Implantation

- Plasma Sheath Surrounds target
- Ions bambard all surfaces of target without beam rastering or target manipulation

In PSII, the ion "beamlets" bombard the target at nearly normal incidence, thereby eliminating the need for target masking.

Figure 2. Plasma Source Ion Implantation (Conrad[12])

As Fig. 2 illustrates, no masking or beam manipulation is required with this method. Plasma source ion implantation provides a uniform dose to all of the target regardless of its shape.[12]

Ion implantation has increased the hardness, corrosion resistance, fatigue resistance, and wear resistance. These four areas are of prime concern to any designer, mechanical engineer, or metallurgist in the design of a tool or component. Certain ions implanted in the heavily used metals such as steel, aluminum, and iron can modify the desired characteristic for that specific usage. Ion implantation increased some of these characteristics from a 100% increase to a 10-fold increase in some cases.[7,10,11]

Ion implantation increases the surface hardness of the target by combining three different reactions between the ions and the target's microstructure. First, the action of the ion slamming into the lattice structure acts as a microscopic peening operation. The ions distort the lattice structure and cause strain hardening to occur in the case of the target. Secondly, the ions travel down the various voids and inclusions to finally wedge in the voids. This action fills the microcracks and boundaries between the grains, increasing the strength of the inter-granular bonds. The third reaction is a chemical reaction that can occur between the target's base metal and the implanted ions. These reactions produce intermetallic compounds such as the formation of iron nitrides in steel and iron, producing a harder alloy on the case of the target.[5,8,13,14]

These increases in hardening are usually done by implanting titanium, nitrogen, boron, argon, and molybdenum. Studies have shown varying success with almost all elements in the periodic table.[15] Nitrogen is used with almost all metals and is used the most often in ion implantation metallurgy. Studies have shown hardness increases of 100% to 200% in the cases of nitride-implanted steels.[7,10,13]

The increased hardness corresponds to a direct increase in the wear resistance of implanted samples. In one case, the wear rates were reduced by up to a factor of 30, others decreased by 1,4, and 10 orders of magnitude.[14,16] The increased hardness is not thought to be the only reason that the wear resistance can be improved dramatically. In some cases, ion implantation lowers the coefficient of friction by a factor of 2, which has a direct influence on the wear rate.[4,7,13]

Ion implantation also decreases wear by encouraging the growth of oxide films that occur between the two sliding surfaces. These films naturally occur on some steels, but the implantation process develops thicker films. This film formation is attributed to the chemical changes brought on by the implantation and the formation of nitrides in the surface layer when implanting with nitrogen.[8,14] Oxide films act as a solid lubricant in the wear mechanism and drastically slow down the wear process.

II. Recent Research

As the amount of research into ion implantation metallurgy increased, the need to start applying this basic research to practical applications became apparent. The vast amount of basic research was starting to be applied with the help of the U.S. Government and the Army Research Office. With the cooperation of the Corpus Christi Army Depot (CCAD) and the Rock Island Arsenal, implanted tooling began to be researched in the metal manufacturing environment. Hughes Research Laboratory expanded their plasma research into the implantation of production tooling used on the shop floor. All three entities have found markedly improved increase in tool life with ion implantation.

The Rock Island Arsenal worked with Dr. John Conrad at the University of Wisconsin-Madison to enhance the wear lifetimes of uncoated and TiN-coated HSS drills. The coated drills were implanted with carbon; uncoated drills were implanted using an acetylene plasma that produced a high hardness-low friction diamond-like carbon (DLC) coating. The TiN-coated and implanted tools showed an increase in tool life and wear resistance, but not as large an improvement as CCAD's or Hughes's tooling. The uncoated tools did not show a statistically significant improvement. Since then, the application of DLC coating has been improved upon, and further research is under way to understand its effect on wear resistance[6]

CCAD compared the performance of nitrogen-implanted tools in a controlled but realistic production environment. They implanted uncoated M4 HSS taps, a TiN-coated tungsten carbide cutting insert, and two uncoated D2 steel alloy punch die sets. After implantation the tools were marked and mixed in with unimplanted tooling, the operators did not know of the difference. The performance criterion was the tool's ability to make parts within specs, the parts made until failure. The taps exhibited a relative performance increase of 1.8 to 4.4; the inserts a 3.8 increase; and the dies, a 1.9 increase.[17]

Hughes Research Laboratories has used plasma ion implantation to achieve a 2 to 3X improvement in wear life of cobalt-cemented, tungsten carbide drill bits, and an 8X reduced wear rate for TiN-coated coated cutting tools. Hughes wanted to validate plasma ion implantation technology in actual field/factory evaluations. Hughes used the same blind test that both Rock Island and CCAD used to test unimplanted tools versus implanted tooling. The experiment was identical to CCAD's to provide a comparison of ion-beam implantation used by CCAD with plasma implantation used at Hughes. [10]

These tests were the first to be done with implanted tooling in a realistic manufacturing environment. They have shown increases in tool life and wear resistance. The small sample size (three tools in Rock Island's study) and uncontrolled shop floor environments show a need for further scientific study into implantation applications. Each study stated a need for further investigation into various tool types, operations, and implantation variables.[18]

III. The Experiment

A. Experimental Design

The experiment had three independent variables of interest: tool type, speed, and feed. The experiment used two tool types, three speeds, and three feeds. A 2 x 3 x 3 factorial design was used to determine the interactions and perform accurate statistical analysis. The 2 x 3 x factorial design does not allow the use of a center point to measure the accuracy of the experiment, so three additional trails at the middle speeds and feeds with each tool type were used. This procedure provided an estimation of the error in the experiment without having to run two or more tests for every level.[19] The 24 trials were randomized using JMP statistical software by SAS.

The experiment involved identical machining parameters for the two types of tooling. The purpose is to machine the work piece long enough to obtain significant wear on the tool to measure the possible decrease in performance associated with the tool wear. The experiment used three different cutting speeds to allow the analysis of different wear rates versus time. The recommended machining parameters for HSS M2 grade tools on 4140 steel are DOC = .04 in., Speed = 135 fpm, Feed = .0070 ipr.[20]

Using this information as a baseline, the speeds and feeds were varied while keeping the depth of cut constant at .05 inches. Three speeds and three feeds were used resulting in nine combinations for each type of tool, plus three extra trials for each tool at the middle speed and feed. The RPM settings on the lathe were the determining factor in the choice of speeds. The feeds were selected by finding the closest feed possible in the lathe to the target feeds of 0.0070, 0.0085, and 0.010. The gearing mechanisms that determine the feed rates allow 0.0069, 0.0084, and 0.0099 respectively.

Speeds	Level	Feeds
125 fpm (160 RPM)	-1	0.0069 ipr
157 fpm (200 RPM)	0	0.0084 ipr
196 fpm (250 RPM)	1	0.0099 ipr

The speeds used are higher than those that are recommended so that the tools would experience accelerated tool wear. At lower speeds, the amount of work piece material and time required would be unreasonable.

Each tool was used for 20 minutes of machining, which allowed for substantial tool wear but was not long enough to induce tool failure. The wear was checked every 5 minutes to predict a wear rate of each tool at the various levels.

The experiment was performed at the New Mexico State University's Advanced Manufacturing Laboratory. A Nardini engine lathe was used to turn the work pieces. The lathe is equipped with a four-jaw chuck and a live center mounted on the tailstock. The work piece was stabilized with a traveling rest and a steady rest.

B. Tools

Twenty four heat treated, M2, right-hand turning, high-speed steel tools with a .075 in. nose radius, 15° end-relief angle, and 44° side-relief angle were used. The composition of M2 tool steel is as follows:

Carbon	1%
Tungsten	6%
Molybdenum	5%
Chromium	4%
Vanadium	2% [21]

High-speed steel tools were chosen because research has shown significant increases in tool life in ion-implanted high-speed steel tools. High-speed steel tools also provide a fast enough wear rate to be measurable within a reasonable amount of time and using a reasonable amount of work piece material. Twelve of the tool bits were sent to Los Alamos National Laboratory to be implanted with nitrogen at the laboratory's plasma source ion implantation facility. They were implanted with nitrogen with retained a dose of 2.5×10^{17} N^+ at/cm$^{2.}$

C. Material

Three 20 ft. lengths of 3-in. diameter, 4140 steel, hot rolled, bar stock, Brinell Hardness - 187[21], was be cut into work pieces 36 in. long. A steady rest and traveling rest was used to eliminate chatter and improve dimensional stability. The length of 36 inches was chosen because that is the maximum length that the lathe can fit when using a tailstock and with the use of the steady rest and traveling rest provides 24 inches of machinable material. 4140 steel is a common alloy that is moderately difficult to machine and can provide significant tool wear.

The composition of 4140 steel is as follows:

Carbon	.38 to .43%
Mn	.75 to 1.00%
P max	.035%
S max	.040%
Silicon	.2 to .35%
Chromium	.8 to 1.1%
Molybdenum	.15 to .25% [21]

D. Data Collection

Data were collected to determine wear rates and surface finish. After each 5-min. machining run, the tool was measured for the resultant wear land. The measurements were done on an Opticom optical magnifier with a 100X magnification power and a measurement accuracy of 0.0005 inches.

Each 5-min. tool run was marked on the bar. When the machinable material on the bar became less than that required for the next trial, the bar was taken off the lathe and five surface roughness measurements were taken on each 5-min. run. The bar was partially rotated after each measurement to preclude any skew in the data. The surface roughness of the resulting work pieces was measured by a SURFCOM 120A surface texture measuring instrument manufactured by Tokyo Seimitsu Co. The Ra (Arithmetic Average), the mean surface roughness, was the variable of interest. The profilometer measured a length of .2953 inches each, with a magnification of 5000X.

IV. Results

The experiment produced four data points per trial for surface finish and tool wear at 5-min. intervals in each test. The data points were averaged to produce a mean wear for each test and divided by the time to produce a wear rate. The results are given in Table I.

Table I: Data for Surface Finish and Tool Wear

		Un-Implanted			Implanted		
Speed	Feed	Mean Wear $x10^{-2}$	Wear Rate $x10^{-5}$ in/min	Surface Finish	Mean Wear $x10^{-2}$	Wear Rate $x10^{-5}$ in/min	Surface Finish
-1	-1	4.56250	9.25	102.05	0.38000	1.25	140.59
-1	0	1.08500	7.00	171.85	0.91625	11.50	184.08
0	-1	1.00875	32.00	173.54	0.94500	11.25	143.83
1	1	1.47250	25.50	237.75	2.01125	5.75	319.57
0	1	5.20625	108.50	253.74	0.73875	9.25	142.72
1	-1	0.76125	9.50	100.86	0.71750	.75	117.88
1	0	1.76875	3.25	197.63	1.49625	4.00	150.43
-1	1	1.28750	6.50	174.92	1.68625	28.25	233.58
0	0	0.95125	12.75	164.18	1.37875	6.50	170.31
0	0	1.27375	10.25	235.43	1.68750	6.25	196.50
0	0	0.88750	14.00	155.15	1.29750	13.75	172.76
0	0	1.19625	14.75	155.08	0.68875	9.75	172.25

The overall relative improvement in tool wear was calculated to be 1.5 times for the implanted tools versus the unimplanted tools. The mean wear for the two tool types at the three speeds is illustrated below in Fig. 3.

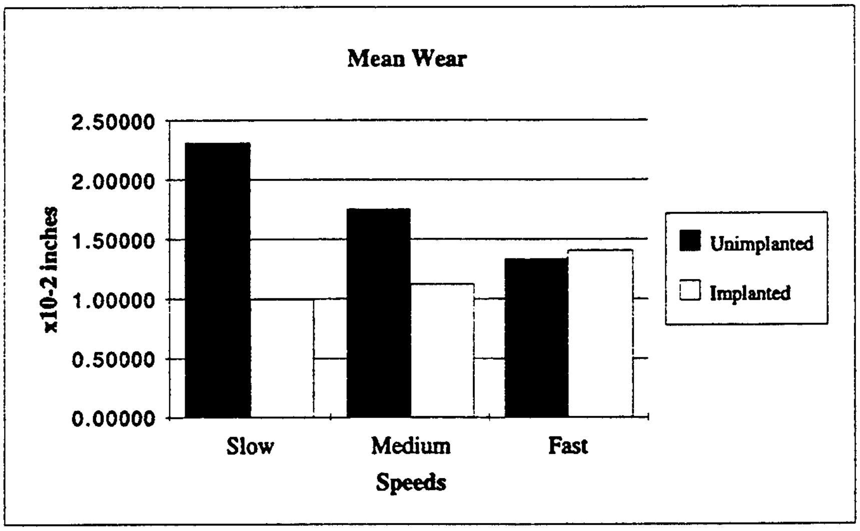


Figure 3. Mean Wear at the 3 Speed Levels

The experiment and mathematical model suggest a four-way analysis of variance (ANOVA). The results from each 5-min. cut and time were used as the variables of interest versus tool wear and surface finish to check for any possible interactions. The JMP software package by SAS was used to produce the results. Tool wear and surface finish were dependent variables in conjunction with independent variables: tool type, speed, feed, and time. The resulting *F* ratios and sum of squares for tool wear is listed in Table II. The results for surface finish are listed in Table III.

Table: II ANOVA results for Tool Wear

Source of Variation	Degrees of Freedom (dF)	Sum of Squares (SS)	F Ratio	Prob>F
Tool Type	1	0.00037950	35.4845	0.0000
Speed	2	0.00074845	34.9914	0.0000
Tool Type * Speed	2	0.00109753	51.3111	0.0000
Feed	2	0.00223420	104.4523	0.0000
Tool Type * Feed	2	0.00051219	23.9457	0.0000
Speed * Feed	4	0.00083297	19.4713	0.0000
Tool Type * Speed * Feed	4	0.00272607	63.7242	0.0000
Time	3	0.00011800	3.6777	0.0261
Tool Type * Time	3	0.00002355	0.7341	0.5419
Speed * Time	6	0.00004860	0.7574	0.6101
Tool Type * Speed * Time	6	0.00004657	0.7257	0.6332
Feed * Time	6	0.00004666	0.7272	0.6321
Tool Type * Feed * Time	6	0.00002006	0.3126	0.9241
Speed * Feed * Time	12	0.00003002	0.2339	0.9941
Tool Type * Speed * Feed * Time	12	0.00005570	0.4340	0.9331

Root Mean Square Error = 0.003270
R Square = 0.969219

Table III: ANOVA results for Surface Finish

Source of Variation	Degrees of Freedom (dF)	Sum of Squares (SS)	F Ratio	Prob>F
Tool Type	1	105.06	0.1031	0.7509
Speed	2	4573.23	2.2449	0.1277
Tool Type * Speed	2	27108.16	13.3071	0.0001
Feed	2	113574.00	55.7522	0.0000
Tool Type * Feed	2	2063.73	1.0131	0.3781
Speed * Feed	4	38211.99	9.3789	0.0001
Tool Type * Speed * Feed	4	38788.72	9.5205	0.0001
Time	3	1718.74	0.5625	0.6450
Tool Type * Time	3	926.02	0.3030	0.8229
Speed * Time	6	3536.28	0.5786	0.7436
Tool Type * Speed * Time	6	6421.75	1.0508	0.4183
Feed * Time	6	1706.57	0.2792	0.9412
Tool Type * Feed * Time	6	3758.25	0.6150	0.7162
Speed * Feed * Time	12	8402.03	0.6874	0.7473
Tool Type * Speed * Feed * Time	12	8099.34	0.6626	0.7688

Root Mean Square Error = 31.91491
R Square = 0.908297

Testing of the hypothesis that each corresponding source of variation has a statistically significant effect on tool wear and surface finish was done by calculating the corresponding *F* statistic and associated p-value (Prob>F). In the analysis of tool wear we find that all the single sources of variation (Tool type, Speed, Feed, and Time) are significant at the 5% level of significance. In addition, the interactions of Tool Type * Speed, Tool Type* Feed , Speed *Feed, and Tool Type *Speed *Feed are also significant.

The effects of speed and feed on tool wear and surface finish have been well documented and studied.[1,19,22,23] These interactions could be expected in an experiment of this type. The high *F* value for Tool Type signifies a statistical difference in the two tool types on tool wear, but the small *F* value for surface finish shows little effect. Implantation does have a significant effect on tool wear but not surface finish.

The resulting r^2 of 0.979219 indicates that there is a strong linear relationship between the variable of interest, tool wear, and the resulting model tested in the ANOVA. The surface finish model showed a strong linear relationship as well with a r^2 of 0.908297.

V. Discussion

The wear rates for the implanted and unimplanted tools are similar at the high speeds in this experiment. The implanted tools performed better than the unimplanted tools at the middle- and lower-speed levels. The researchers believe that these similarities can be expected because of the nature of ion implantation. The similarities of the tools at the highest speed level are attributed to the high temperatures that are present in the tool work piece interface. The high temperatures precipitate thermal softening at the tip of the tool. Thermal softening is the common mode of failure in high-speed steel tools.[23] The implantation effects have been known to dissipate quickly at high temperatures.

Ion implantation's effect on surface hardness and wear resistance has been well studied in the laboratory environment. This study shows that ion implantation does have a significant effect on improving wear resistance of high-speed steel tools when implanted with nitrogen. Ion implantation does not have a sizable effect on performance as measured by the resulting surface finish of the work piece.

Further research needs to be done on effects that implantation has on various cutting tools. Many avenues still need to be explored. Because, implantation does effect the tool's wear resistance, the new tool needs to have recommended speeds and feeds developed for it to fully utilize its increased performance. These studies could be expanded to include implanting other tool types (carbides, coated carbides, TiN-coated tools) and their corresponding wear improvements to develop corresponding tool life equations and recommended processing parameters.

The basic research to this point involved understanding the effects of changing voltages, fluences, dopants, and other implantation variables to find an optimum hardness improvement in the material. This research should be expanded to include tooling to determine the optimum implantation parameters. This study used nitrogen at 45 keV because that amount had been determined to be the optimum level to increase hardness in tool steel. The increase in hardness may not result an increase in wear resistance and should be studied further.

VI. References

1. Lambert, B. K., Advanced Manufacturing Processes, I.E. Course, New Mexico State University, Fall 1994.

2. DeGarmo, E. P., J.T. Black, and R. A. Kohser, *Materials and Processes in Manufacturing*, MacMillan Publishing Co., New York, 1984.

3. Hunt, Gordon, *Improving the Performance of Cutting Tools and Dies by Ion Assisted TiN Coatings*, unpublished, 1993.

4. Townsend, P. D., J.C. Kelly, N. E. W. Hartley, *Ion Implantation, Sputtering and Their Applications*, Academic Press, Orlando, 1976.

5. Scheuer, J. T., K.C. Walter, J. P. Blanchard, D. J. Rej, and M. Nastasi, *Plasma Source Ion Implantation of Ammonia into Electroplated Chromium*, TMS Annual Meeting, 1995.

6. Conrad, J. R., K. Sridharan, R. P. Fetherston, A. Chen, M. M. Shamim, J. Firmiss, *Carbon Plasma Source Ion Imlantation/Deposition of Drills Used at Rock Island Arsenal*, unpublished, 1994.

7. Rej, D. J., *Large-scale Plasma Source Ion Implantation Program for Improving Mechanical Properties of Automotive Parts and Tooling*, Materials Technology, 8:85-93, 1993.

8. Rej, D. J., J. R. Conrad, R. J. Faehl. R. J. Gribble, I. Henins, N. Horswill, P. Kodalli, M. Nastsi, W. A. Reass, M. Shamim, K. Sridharan, J. R. Tesmer, K. C. Walter, and B. P. Wood, *First Results from the Los Alamos Plasma Source Ion Implantation Experiment*, Material Research Symposium Proceedings. Vol. 316, 1994.

9. Reeber, R. G. Dearneley, G. R. Alexander, K. Legg, J. Moriarity, J. Mayer, G. Collins, R. Gunzel, *Plasma Source Ion Implantation 1993 - A Critical Assessment*, The First Annual Workshop of Plasma Source Ion Implantation, 1993.

10. Conrad, J. R., K. Sridharan, M. M. Shamim, R. P. Fetherston, A. Chen, A. A. Elmoursi, A. H. Hamdi, G. W. Malaczynski, X. Qiu, R. J. Faehl, M. Nastasi, W. A. Reass, D.J. Rej, J. T. Scheuer, B. P. Wood, *Plasma Source Ion Implantation: An Environmentatly Acceptable Alternative to Wet Chemical Plating,* International Journal of Environmentally Conscious Design & Manufacturing, Vol. 2, No. 1, pp. 61-66, 1993.

11. Meletis, E.I., S. Yan, *Low-pressure Ion Nitriding of AISI 304 Austentic Stainless Steel with an Intensified Glow Discharge,* Journal of Vacuum Science Technology, Jan/Feb 1993.

12. Conrad, J. R. , S, Baumann, R. Fleming, and G.P. Meeker, *Plasma Source Ion Implantation Dose Uniformity of a 2x2 Array of Spherical Targets*, Journal of Applied Physics, Vol. 65, February 15, 1989.

13. Malik, S. M., K. Sridharan, R. P. Fetherston, A. Chen, J.R. Conrad, *Overview of Plasma Source Ion Implantation Research at University of Wisconsin - Madison*, Journal Of Vacuum Science Technology, Mar/Apr, 1994.

14. Reinbold, Rolland A., Ronald A. Kohser, Edward B. Hale, *Wear Improvements Induced by Thermally Grown Oxide Layers and by Nitrogen Ion Implantation*, Journal of Applied Physics, August 15, 1991.

15. Brown, Ian G., *Metal Ion Implantation for Large Scale Surface Modification*, Journal of Vacuum Science Technology, Jul/Aug 1993.

16. Reuther, H., B. Rauschenbach, E. Richter, *Ion Implantation in Metals-Structure Investigations and Applications*, Vacuum, 1988.

17. Culbertson, R.J., F.C. Burns, W. Franzen, L.J. Lowder, J.J. Ricca, A. Gonzales, *Performance Evaluation of Ion-Implanter Cutting Tools and Dies,* Nuclear Intruments and Methods in Physics Research, B56/57, 652-655, 1991.

18. Matossian, J. N., *Plasma Source Implantation Technology at Hughes Research Laboratories*, Journal of Vacuum Science Technology, Mar/Apr, 1994.

19. Lambert, B. K., *Determination of Metal Removal Rate with Surface Finish Restriction*, First Annual Material Removal Conference, Society of Manufacturing Engineers, 1983.

20. Machinability Data Center, *Machining Data Handbook*, Metcut Research Associates Inc., 1980.

21. Society of Manufacturing Engineers, *Tool and Manufacturing Engineers Handbook*, McGraw Hill, New York, 1976.

22. Boothroyd, Geoffrey, *Fundamentals of Metal Machining and Machine Tools*, Scripta Book Co., Washington, D.C., 1975.

23. Venkatesh, V.C., *A Discussion on Tool Life Criteria for HSS, Carbides and Oxide Ceramics*, Society of Manufacturing Engineers International Conference and Exposition, 1989.

TASK 3.4

PUBLICATIONS AND TECHNICAL REPORTS PUBLISHED

MACHINING STRATEGIES COLLECTED, STORED AND UTILIZED WITH ARTIFICIAL NEURAL NETWORKS

LEON D. COX, AMJED AL-GHANIM, & DAVID E. CULLER
New Mexico State University, Integrated Manufacturing Systems Lab
Box 30001 Dept. 4230, Las Cruces, NM 88003 (lcox@nmsu.edu)

ABSTRACT:
This paper presents results from an Army Research Office grant developing knowledge based systems for Computer Numerical Control (CNC) machining operations. One of the areas of development involves knowledge/data acquisition for input parameters to the knowledge processor. In our work we chose artificial neural networks as a mechanism for capturing data and then utilized the ability of the networks to determine certain process parameters for automating specific tasks in CNC programming. Artificial neural networks are well adapted to this task because they can be continuously improved.

INTRODUCTION

An increasing need for the rapid production of small batches and prototype parts will drive manufacturers to seek new ways to decrease lead times and costs associated with Computer Numerical Control (CNC) programming and production planning. Planning and developing CNC programs for the production of mechanical parts can be an expensive and time-consuming process. A process engineer (domain expert), who traditionally performs these functions routinely applies learned skills to prescribe manufacturing operation steps and specific machining parameters. One author summarizes the problem as follows:

> This knowledge is better described as many differently weighted pieces of advice than as a small set of constraints to be satisfied. Such planning involves taking into account a great variety of both technological rules and economic considerations. Furthermore, they are not absolute. They are more preferences, between which compromises may be necessary. In addition, they may differ somewhat from one company to another. In fact, they represent the experience and know-how of engineers. An immediate consequence is that the planning of machining sequences definitely does not have a unique solution. (Descotte and Latombe, 1989)

Problem Identification

Commercial CAD/CAM/CAPP software do not currently facilitate the automatic accumulation of design and manufacturing data produced throughout the planning and production of parts. Today's CNC machine tool programming methods are often inefficient and do not utilize the product model concept. Plans are often produced and executed on the fly, modified on only the final hard copy, and inaccessible by people responsible for job costing. Furthermore, the intelligence and know-how to manufacture and track parts resides in a wide base of skilled crafts-people, process engineers and manufacturing experts.

Commercial CAD/CAM gets limited acceptance among CNC machine operators and CNC programmers who perceive that their jobs are being replaced by computers. And of course the systems available for efficiently capturing the knowledge of the 'skilled' are for the most part still in a state of development.

Artificial Neural Networks

There has been an increase of interest in 'brain style computing' in terms of artificial neural networks (ANNs). ANNs are parallel distributed processing architectures composed of a huge number of small interacting elements that are massively interconnected. Each of these elements sends excitatory and inhibitory signals to other units that in turn update their behaviors based on these received messages. ANNs emulate the functionality of the human brain that implicitly stores knowledge in the interconnection weights, and not in the neurons themselves (Rumelhart and McClelland, 1986). Learning is thus achieved by modifying the connection weights between elements. Feedforward neural nets acquire knowledge through training, where by repeatedly presenting sample cases to the net, its interconnection weights change accordingly to model the representation of the cases (Shalkof, 1992). Knowledge is stored in the final interconnections values. As such neural network methodology would ease the knowledge acquisition bottleneck that is hampering the creation of expert system (Zurada, 1992). The key to a neural network's power is the inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization. A typical network consists of a set of processing elements (PE) grouped hierarchically in layers and interconnected in some fashion. These PEs sum N weighted inputs and transfer that outcome through a mathematical function. One of the most popular neural paradigms, and the one that has been used in this project, is the multilayer perceptron trained using the Error Back Propagation Training Algorithm (EBPTA) (Dayhoff, 1990). Within the manufacturing environment ANNs have been applied to quality control, process control, part-family classification (Martinez, Smith, and Bidanda, 1993; Telasco, 1993; Udo, 1992; Cook and Shannon, 1992; Charraprty and Uptal, 1993) Neural network models have been also used for optimizing cutting parameters of turning operations (Wang, 1993).

Neural network models have key advantages over rule-based systems. First, the learning process, or the knowledge acquisition process, takes place by presenting only sample (machining) cases to the network. Learning occurs through modifying the connection weights between network elements (by a learning algorithm) to model the representation of the cases (Shalkof, 1992). Knowledge is thus stored implicitly in the final interconnection values. Second, because training examples need not be exhaustive, it is conceivable that neural network methodology could ease the knowledge acquisition bottleneck that is hampering the creation of rule-based systems (Zurada, 1992). Finally, when utilized as a decision processor, an artificial neural network has inherent parallelism resulting in fast computation, robustness or error resistance, and adaptiveness or generalization.

Computer Aided Process Planning

Computer Aided Process and operations Planning (CAPP) systems attempt to automate the decision making process necessary to map design specifications on the company specific manufacturing environment. Many CAPP systems utilize some form of Artificial Intelligence (AI) to model the human interaction previously required by the task. CAD/CAM software and databases are used to supply product descriptions and information management, respectively. Material removal operations provide a rich environment for the implementation of intelligent applications. Much of the current work in this field concentrates on the AI technique used to process the knowledge and/or the solid model design features utilized by the engineer to initially create the part (Anderson and Chang, 1990; Chung, Patel, and Cook, 1990; Joshi, Vissa, and Chang, 1989; Phillips and Mouleeswaran, 1985).

Manufacturing Assistants

Manufacturing Assistants are systems which emphasize the specific decisions and associated data which must be considered when selecting tools, cutting speeds, tool motions, etc. in a manufacturing or machining environment. As discussed here, the term machining is defined as *the removal of unwanted material from a workpiece in chip form so as to obtain a completed product that meets size, shape and finish characteristics specified by the designer.* The following factors contribute to the need for improved methods to manage the manufacture of machined parts: advanced machine tool and computer technology, higher quality and stricter tolerance and finish requirements, new materials such as plastics, composites, and exotic alloys, smaller batch sizes and prototype parts, more complex part descriptions, and a more competitive marketplace.

The Turning Assistant

The Turning Assistant (TA) is a direct outcome of an ARO project which automatically creates plans for Computer Numerical Control (CNC) lathe operations. Starting with a drawing and a set of predefined machines and cutting tools, the TA interactively defines and produces code for a variety of features such as profiles, grooves, and threads. The TA focuses on the formalization of domain knowledge, providing document support, and reducing the costs and lead times associated with the planning function. For advanced systems to be usable in shops which currently dominate small batch production of machined parts, flexible tools for fitting the application to the environment must be made available.

The Turning Assistant is a system that is based on previous work providing automated programming for a variety of milled features. The "Milling Assistant" (MA) (Burd, 1989) evolved from an Expert System based decision processor to a Case-Based Reasoning (CBR) system to automate the decision process required for machining milled and point-to-point features on CNC machine tools.

PRODUCT MODELS

The complete electronic product definition for a manufactured part consists of all the information related to designing, producing, inspecting, and managing the manufacture of that part. The data provided by the product model must be accessible, well organized, and conform to standards prevalent in both industry and government. The Computer

Aided Acquisition and Logistics Support (CALS) and Product Definition Exchange Specification (PDES) / Standard for the Exchange of Product Data (STEP) initiatives are currently addressing these very issues.

Knowledge processor

Numerous systems can be utilized for decision processing. The TA decision processor is written in GRAPL-IV, a FORTRAN 77 based application interface within the ANVIL-5000 CAD/CAM system and is used to determine machining strategies and effectively execute the operation. Another is based on a series of ANN's written in C, which can be called as needed from the application interface within the ANVIL-5000 CAD/CAM system. Another set of C programs can make calls to MetCAPP-III or MetCAPP-IV to acquire necessary machining parameters and cutting strategies.

KNOWLEDGE ACQUISITION AND REPRESENTATION

The most important and difficult aspect of this research is to develop effective tools with which to elicit knowledge from domain experts. The skills obtained through years of observation, training, improvising, and trial-and-error are not easy to explain verbally, document, or formalize for the purpose of building automated CNC programming systems. A significant amount of time must be spent in familiarizing oneself with the field and terminology before attempting to solicit strategies directly from the CNC programmers and machinists.

ANN's have a distinct advantage in speed, development and knowledge updating (hence learning). Automatic creation of CNC programs requires: 1) user-interaction -- to gather location specific operating information, 2) interrogation of the features geometric description within the CAD database, 3) a file based communication link to a decision processor and, 4) An applications interface language capable of building tool paths from user generated programs. Very few commercial CAD/CAM systems provide the user with enough power to accomplish these tasks.

Representing the Knowledge

This research helped to develop knowledge elicitation and representation tools that can be used to formalize the expertise gained through machining strategy meetings, literature review, and programmer feedback. It includes the design of questionnaires that can be used to document strategies in a set format during interviews or literature review and data collection forms for training ANN's (see Table 1).

TABLE 1: ANN TOOL TYPE DATA COLLECTION FORM

Input Decision Variables									Output Decision Variables
*	Variable Name	Value	Weight Factor						The correct tool material:
1	Part (stock) material		0	2	4	6	8	10	1. High Speed Steel
2	Part material bH		0	2	4	6	8	10	2. Carbide
3	Material Machinability		0	24	-462	6	8	10	3. Ceramic
4	Feature Depth		0	2	4	6	8	10	Comments:

Artificial Neural Networks as Knowledge Acquisition Tools

The adopted research methodology aimed at investigating the utilization of a neural-based knowledge acquisition approach. It is based on the operational mode of neural systems, and taking advantage of our knowledge and experience in developing rule-based systems for similar tasks (i.e. milling assistant (Burd, 1990)). This methodology satisfies a number of basic requirements. First, it must reduce the burden associated with developing rule-based systems by avoiding constructing general rules, because, as explained earlier, experts are not good at generalizing solutions. Second, this methodology must ensure that knowledge be cast in a form usable by neural systems; forms of rules are excluded. What is required by a neural system is a set of training examples, where each example represents a real machining case described by a vector of numeric values of process variables encoded in a convenient from. Third, consequently, the new methodology must employ effective instrumental knowledge gathering forms; and since no general inferential statements are required at this stage, the knowledge gathering forms can be mere data collection forms

A neural network paradigm must first be selected in consensus with the application at hand. In this research, a *supervised* neural network model is employed as a *classifier*. A supervised learning model is selected as it enables the expert to express his judgmental opinion in a relative manner. This feature is very essential because, in this application, it is not required to group the training examples into clusters as would be the outcome if an *unsupervised* learning model were used. Instead, every single training example must be explicitly identified (by the expert) as belonging to a specific class that represents a machining decision. The neural model is employed as a classifier so that the output decision belongs to a class of objects (e.g. type of tool material: HSS, Carbide, Ceramic), or to a range of numeric values (e.g. tool size: 1/4"-1/2", 1/2"-1", etc.). This method shows great promise considering the variety of machining choices that can satisfy the requirements of a quality product. See Table 2 for a representative example of the ANN training and testing results.

ACKNOWLEDGMENTS

The members of this project are grateful to Army Research Office, Boeing Corp., Ladish Corp., Manufacturing and Consulting Services, Institute for Advanced Manufacturing Sciences, Physical Sciences Laboratory and Sandia National Laboratories for their contributions of grants, expertise, equipment and software.

TABLE 2: TOOL MATERIAL NETWORK CLASSIFICATION PERFORMANCE

Training Examples			Testing Examples		
hidden nodes	learning constant(η)	identification correct (%)	hidden nodes	learning constant(η)	identification correct (%)
5	.20	90	5	.20	80
11	.20	95	11	.20	85
16	.20	97	16	.20	90

REFERENCES

Anderson, D.C., and Chang, T.C., (1990). Geometric reasoning in feature-based design and process planning. *Comput. & Graphics*, Vol. 14, No. 2, pp. 225-235.

Burd, W.C., (1989). The Milling Assistant, *SAND89-1519, Computer Aided Manufacturing Division*, 7483, Sandia National Laboratories, Albuquerque, New Mexico, 1989.

Charraprty, Kanda, and Utpal, Roy, (1993). Connectionist model for part family classification, *Computers and Industrial Eng.*, Vol.24, No.2, pp.189-198.

Chung, J.C.H., Patel, D.R., & Cook, R.L., (1990). Feature-based modeling for mechanical design, *Comput. & Graphics*, Vol. 14, No. 2, pp. 189-199.

Cook, Deborah F., and Shannon, Robert F., (1992). A predictive neural network modeling system for manufacturing process parameters, *International Journal of Production Research*, Vol.30, No.7, pp.1537-1550.

Dayhoff J.E., (1990) *Neural networks architectures: An introduction*, van Nostrand International Company, New York.

Descotte, Y., and Latombe, J.C., (1989). Gari: A problem solver that plans how to machine mechanical parts, *I.M.A.G.*, Grenoble Cedex, France.

Gallant, Stephen, (1993). Neural Network Learning and Expert Systems, MIT press.

Joshi, Vissa, & Chang, (1989). Expert process planning system with a solid model interface, Chapter 6, *Knowledge-Based Systems in Manufacturing*, Taylor and Francis Inc. Philadelphia, PA.

Martinez, S., Smith, A., Bidanda, B. (1993). A neural predictive quality model for slip casting using categorical metrics, *Proc. of IIE Conf.*, Los Angeles, May 26, 1993, pp.265-269.

Phillips, R.H., and Mouleeswaran, C.B., (1985). A knowledge-based approach to generative process planning, *CASA/SME Autofact 85 Conference*, November 1985.

Rumelhart D.E., McClelland J.L., & the PDP Research Group, (1986). *Parallel distributed processing: Explorations in the microstructure cognition*, Volume 1, MIT Press.

Shalkof R.J., (1992). *Pattern recognition: Statistical, structural, and neural approaches*, John Wiley & Sons, Inc., USA.

Telsaco, Thomas, (1993). Interfacing process quality control and CIM through application of artificial neural networks, *Proc. of IIE Conference*, Los Angeles, May 26, 1993, pp.280-284.

Udo, Godwin J., (1992). Neural networks applications in manufacturing processes, *Computers and Industrial Eng.*, Vol.23, No.1-4, pp.97-100.

Wang, Jun, (1993). A neural network approach to multiple-objective cutting parameter optimization based on fuzzy preference information, *Computers and Industrial Engineering*, Nos 1-4, pp.389-392.

Zurada J.M., (1992) *Introduction to artificial neural systems*, West Publishing Company, USA.

TASK 3.4

PARTICIPATING SCIENTIFIC PERSONNEL

TASK 3.4 PARTICPATING SCIENTIFIC PERSONNEL

NAME	STATUS
Prof. J. H. Bentley	Professor
Prof. L. D. Cox	Associate Professor
John D. Holder	BS Awarded
Dr. B. K. Lambert	Professor
Eric D. Parsons	MS Awarded
Ralph Williams	BS Awarded
Roger H. Woods	MS Awarded

LASER MONITORING, DIAGNOSIS, AND CONTROL

Thomas Shay

Department of Mechanical Engineering
New Mexico State University

Final Report for Task 3.5 of Applied Manufacturing Research
Submitted to the Army Research Office

February 1996

Laser Metrology Research
Task 3.5 Final Report

EXECUTIVE SUMMARY

Accurate real time displacement measurements are essential to increasing the productivity and decreasing the cost of ultra-high precision machining. This research presents a first step in that direction. We have demonstrated a simple measurement system that can operate with the machine cutting and maintain an measurement accuracy of 0.2 micro-inches. We used the sinusoidal phase measurement technique using laser diodes, SPM-LD, and have invented a new algorithm for data analysis that allows for the first time the extension of this method to dynamic ranges on the order of ten inches with high accuracy. We have demonstrated a dynamic range that is a factor of 1,000 greater than the previous research. However, our demonstrated dynamic range was limited by our hardware to 0.04 inches, better hardware will easily increase the dynamic range to several inches at high accuracy. In general, the dynamic range can be increased if we relax the accuracy requirements. Finally, an invention disclosure is being filed on this measurement technique. The original goal of this effort was to demonstrate a system that had a dynamic range of ½ inch and a resolution of 20 micro-inches in a vibrating environment, as you can see we have clearly exceeded the original goals. The next step in this research would be to build a prototype system to place on a diamond turning machine.

This work was the MS thesis of Mr. Chi Man and he was supported by this grant. Also supported in this grant were Mr. Lawrence Alvarez, a staff engineer at NMSU, Prof. Yi Fang Wu, a visiting professor from Bejing University, Dr. Bing Yin and Professor T. M. Shay of New Mexico State University.

PUBLICATIONS

May 95, MS, Thesis, Man, Chi K., "High-Resolution Displacement Measurement by Harmonic Demodulation in SPM-LD Interferometry"

CLEO'97, Conference paper accepted, Y. F. Wu, C. K. Man, B. Yin, M. K. Giles, and T. M. Shay, "5 Nanometer Resolution in a Vibrating Environment with a 1 mm dynamic range"

1. Introduction

Our purpose is to develop a real-time measurement displacement system with mm order dynamic range and 5 nm resolution. Thus, we are able to obtain a wide dynamic range and high resolution in a simple interferometer that can tolerate vibration of up to 200 nm. This system can provide the crucial part of a closed loop for precision machine tools.

As is well known, interferometers are theoretically capable of wide dynamic range and high resolution. The resolution is theoretically limited only by the quantum noise (shot noise) from a photodetector. It is difficult to implement both wide dynamic range and high resolution simultaneously in a single experimental system, because high resolution depends upon how a small fraction of the wavelength can be resolved and dynamic range depends on how many fringes are counted reliably.

Several techniques have been used to overcome these drawbacks: two wavelength interferometry[2], frequency modulated continuous wave LD interferometry[3] ect. In Ref. 3 the wavelength is linearly ramped and that system demonstrated a range of between 5 - 600 mm with a resolution of 50 μm, the resolution is too low for precision machining applications. In References 4 and 5, the Sinusoidal Phase Modulation (SPM) method is used, there the wavelength is modulated with a sinusoidal wave and the phase of the interfering signal depends upon the path difference of the interferometer legs. The phase of interference signal is calculated using a synchronous detection to obtain the amplitude of the first and second harmonic components of the SPM frequency. They demonstrated a dynamic range of λ and measure a resolution of λ/800 and show that a range of $2^3 \bullet \lambda$ and a resolution of λ/255 should be possible. The range of there system was limited by the requirement that the amplitudes of the first and second harmonic detection loops be appropriately balanced. The required ratio of the gain for the first and second harmonic loops changes as the displacement changes and hence using their algorithm the dynamic range must be kept short or the accuracy falls off rapidly. Furthermore, as the gain of the circuit is changed the relative phase shift between the 2 circuits will change, since we are measuring a phase change this introduces a systemic error.

In this work, we developed the theory of Wide Dynamic Range Sinusoidal Phase Modulation Laser Diode (WDR-SPM-LD) interferometry which uses a slowly-varying approximation and we present a new algorithm for calculating the phase of the SPM signal. The advantages of this new algorithm are: first, it is significantly less sensitive to changes in the amplitudes of the first or second harmonic and hereby increases the dynamic range by 3 orders of magnitude; second, the simplicity of the phase demodulating circuit is preserved; finally, it can be implemented on a computer for a real-time automatic measurement displacement. Thus, our system can be implemented in a closed loop wide dynamic range system for precision machining. Our system consists of two unbalanced Michelson interferometers, a harmonic phase demodulator, an active feedback control system that compensates for thermal drifts, as well as, reduces the effects of external environmental disturbances, 12-bit A/D conversion and automatic real-time measurement displacement in LabVIEW software by a computer. We used both a PZT and a Picomotor as the displacement actuators for measuring the resolution or a dynamic range.

Experimental results show that 1 mm range and $\frac{2\pi}{2^{12}}$ phase resolution are achieved. The phase resolution corresponds to a 0.2 nm spatial resolution. The dynamic range has been increased by over 3 orders of magnitude compared to the previous state of the art[5]. The absolute accuracy of the measurements depends upon the reproducibility of the actuator itself. The New Focus Picomotor was used for the wide dynamic range measurements and as stated by New Focus the positioning precision is 0.1 %, this was calibrated against a HeindenHain gauge and compared to the measurements of our system. The accuracy of our measurements agreed with the measurements by the Heindenhain gauge. Using WDR-SPM-LD we were able to calibrate our Picomotor.

2. Theory

2.1 Laser Diode Sinusoidal phase modulating (SPM) interferometer

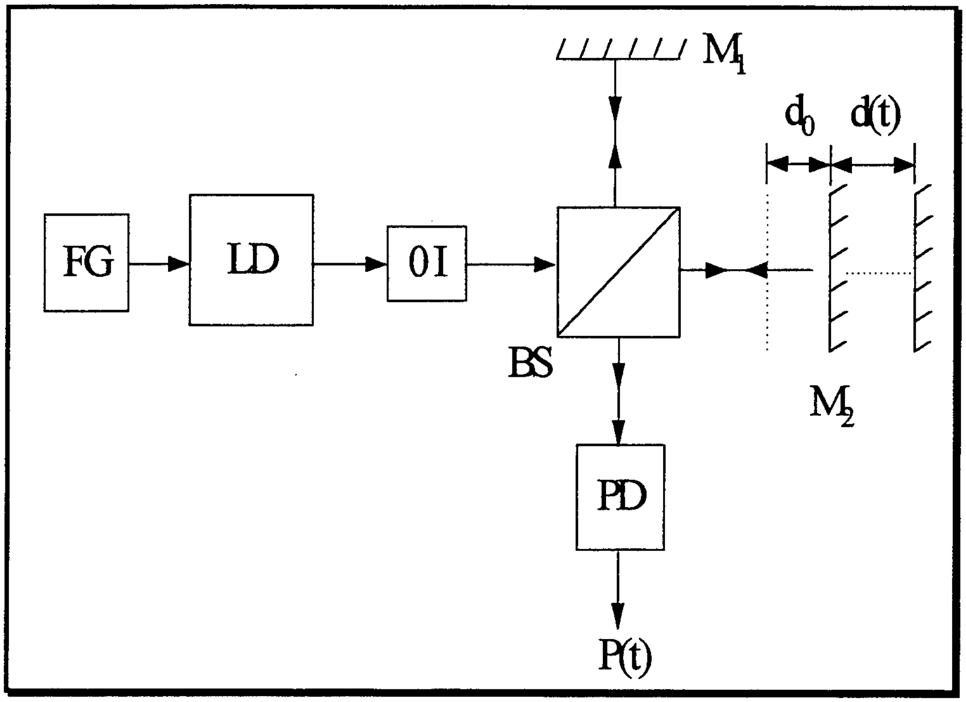


Fig. 1a Block diagram of the SPM-LD interferometry system.

Figure 1a shows the setup of SPM-LD interferometer for real-time displacement measurement. The optical configuration is an unbalanced Michelson interferometer. The light emitted from the laser diode (LD) passes through an optical isolator (OI) to prevent optical feedback and is split into a reference beam and an object beam with a beam splitter (BS). The optical path difference between the two arms of the interferometer is $\ell = 2x$.

The injection current of the LD consists of a dc bias current i_0 and modulation current $i_m(t)$, which are supplied by the laser diode controller (LC). Current i_0 determines the central wavelength the λ_0 of the LD. The modulation current $i_m(t)$ is

$$i_m(t) = a \cos\left(\omega_m t\right) \tag{1}$$

Because the modulation current changes the wavelength by $\beta i_m(t)$, sinusoidal phase modulation is achieved, where β is the modulation efficiency. The interference signal is detected by a photodetector, PD. The time-varying component, or ac component, of interference is given by

$$P(t) = P_0 \cos\left[z \cos \omega_m t \; + \; \theta\right] \tag{2}$$

where z = $4\pi a \beta x \Big/ \lambda_0^2$, $\theta = \dfrac{4\pi}{\lambda_0}\chi$ and χ is the distance difference between the two arms of the interferometer, P_0 is the amplitude of the ac component.

2.2 The Harmonic Phase Demodulator

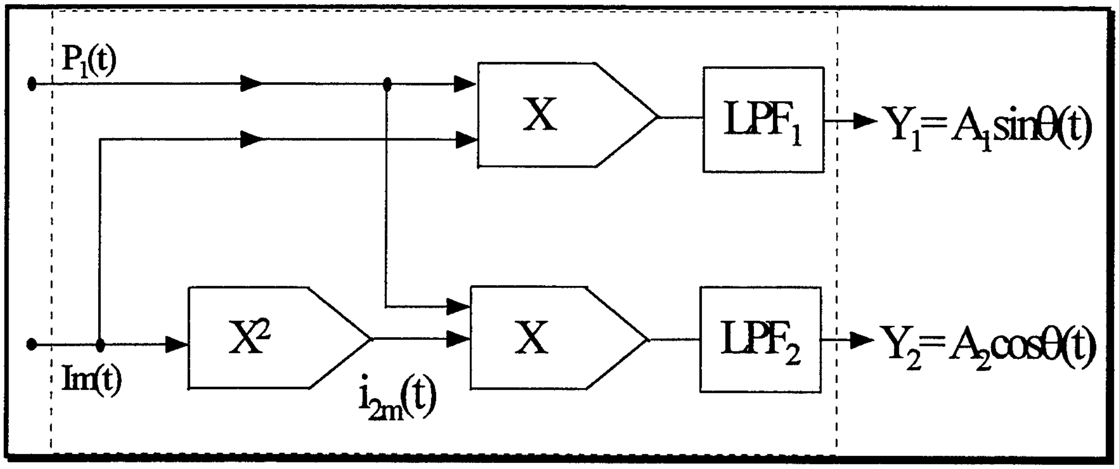


Fig. 1b Shows a block diagram of the harmonic demodulation system.

In equation (2), the unknown displacement x is obtained from the phase θ, but θ has an ambiguity of 2π. The key task is accurately and unambiguously to extract the interferometric phase. For this purpose the demodulator is required to have a high resolution and a wide dynamic range. In this article, we use multi-harmonic phase demodulation, that is, the interference signal P(t) is synchronously demodulated by multiplying it times $i_m(t)$ and integrating to obtain the amplitude of the quadrature components, or sin θ and cos θ. The block diagram is shown in Fig. 1b. Expanding Eq. (2), we obtain

$$P(t) = P_0 \cos \theta[J_o(z) - 2J_2(z) \cos 2\,\omega_m t + \ldots]$$
$$-P_0 \sin \theta\left[2J_1(z) \cos \omega_m t - 2J_3(z) \cos 3\omega_m t + \ldots\right] \tag{3}$$

where $J_n(z)$ is an n'th-order Bessel function. Multiplying $P(t)$ by $i_m(t)$ and passing it through the low-pass filter (LPF), we obtain the signal associated with the first harmonic frequency component as follows:

$$Y_1 = A_1 \sin \theta \tag{4}$$

where

$$A_1 = -2K_1\, P_o a J_1(z) \tag{5}$$

where K_1 is the LPF_1 gain. Squaring $i_m(t)$ and suppressing the dc component, we obtain the signal

$$i_{m2}(t) = \left(\frac{a^2}{2}\right) \cos\left(2\omega_m t\right) \tag{6}$$

Multiplying P(t) by $i_{m2}(t)$ and passing it through the low-pass filter (LPF_2), the desired signal associated with the second frequency component can be obtained as follows:

$$Y_2 = A_2 \cos \theta \tag{7}$$

where

$$A_2 = -K_2 P_0 a^2 J_2(z) \tag{8}$$

where K_2 is the LPF_2 gain. The signal processing to obtain the signal of Eq. (4) and (7) are called harmonic demodulator. Combined with Eq. (4) and (7), the phase θ can be calculated from

$$\theta = \tan^{-1} \frac{\left(Y_1 / A_1\right)}{\left(Y_2 / A_2\right)} \tag{9}$$

Thus, once θ and λ are known then the displacement can be calculated using,

$$\theta = \frac{2\pi}{\lambda_0} * \chi$$

2.3 Slowly-varying Approximation

If, as in Reference [5], K$_1$ and K$_2$ were set to satisfy the condition, A$_1$=A$_2$, then Eq. (9) becomes

$$\theta = \tan^{-1}\left(\frac{Y_1}{Y_2}\right) \qquad . \qquad (10)$$

The authors of Ref.[5] used Eq. (10) to calculate the interference phase , then displacement. However, due to the limited gain range of LPF, especially, the unbalance of the electronic circuits and imperfection of the electrical components, the additional errors due to phase shifts of the electronic circuits occur when the gain of LPF is adjusted. Thus, it is difficult to increase the dynamic range beyond $\left(2^4 \bullet \frac{\lambda}{2}\right)$.

Now, we present a new principle that provides a much wider dynamic range displacement measurement and avoids any adjustments in the circuit gain.
In contrast to the assumptions required inorder to use Eq. 10 the amplitudes A$_1$ and A$_2$ change as a function of χ and thus Eq. 10 is not valid for large dynamic ranges. We develop the slowly-varying approximation theory that automatically compensates for changes of the amplitudes A$_1$ and A$_2$. The theory is summarized below.

Since the change of the amplitudes A$_1$ and A$_2$ vary extremely slowly over distances on the order of λ and a single step of the positioning device is limited to less than $\lambda/2$. Thus we can use can use the slowly-varying approximation, that is, $\frac{dA_1(x)}{dx} \approx 0$ and $\frac{dA_2(x)}{dx} \approx 0$.

Differentiating Eqs. (4) and (5). We can obtain

$$\frac{dY_1}{dx} = A_1(2K)\cos(2\mathrm{K}x) \qquad (11)$$

$$\frac{dY_2}{dx} = A_2(-2K)\sin(2\mathrm{K}x) \qquad (12)$$

Combining Eqs. (4), (7) and (11), (12), we get

$$\tan^2\theta = -\frac{Y_1}{Y_2}\frac{\left(\frac{dY_2}{dx}\right)}{\left(\frac{dY_1}{dx}\right)} \qquad (13)$$

the phase θ is calculated from

$$\theta = \tan^{-1}\left(\left(-1\right)\frac{Y_1}{Y_2}\cdot\frac{\left(\frac{dY_2}{dx}\right)}{\left(\frac{dY_1}{dx}\right)}\right)^{\frac{1}{2}} \qquad (14)$$

Considering the signs of Y_1, Y_2, $\frac{dY_1}{dx}$ and $\frac{dY_2}{dx}$ in four quadrants, it is not difficult to understand the meaning of the minus of square root in the right of Eq. (14). From Eq.(14),we developed a novel method to calculate an interference phase, in which an affecting to Y_1 (or Y_2) due to the change of amplitudes A_1 (or A_2) itself has been considered and determined by using $\frac{dY_1}{dx}$ and $\frac{dY_2}{dx}$ in SPM LD interferometry.

3. The Feedback Control System to Reduce the External Disturbances

Wavelength modulating is easy to implement with a laser diode since its output wavelength varies linearly with the injection current over most of its operating range. However, the wavelength fluctuation $\Delta\lambda_{LD}$ due to changes in the operating conditions has to be overcome if high precision measurements are to be made.

In addition, optical devices in an interferometer vibrate in response to external mechanical, acoustic vibrations. Moreover, the interferometer components undergo thermal expansion due to fluctuations in temperature. All of these effects cause the changes $\Delta\ell$ in the optical path difference ℓ between the object and reference waves. These $\Delta\lambda_{LD}$ and $\Delta\ell$ cause the fluctuation in the phase of the interference signal. The fluctuation can be reduced by controlling the injection current to produce the change $\Delta\lambda_f$ in the wavelength of LD, as reported in Ref. [4].

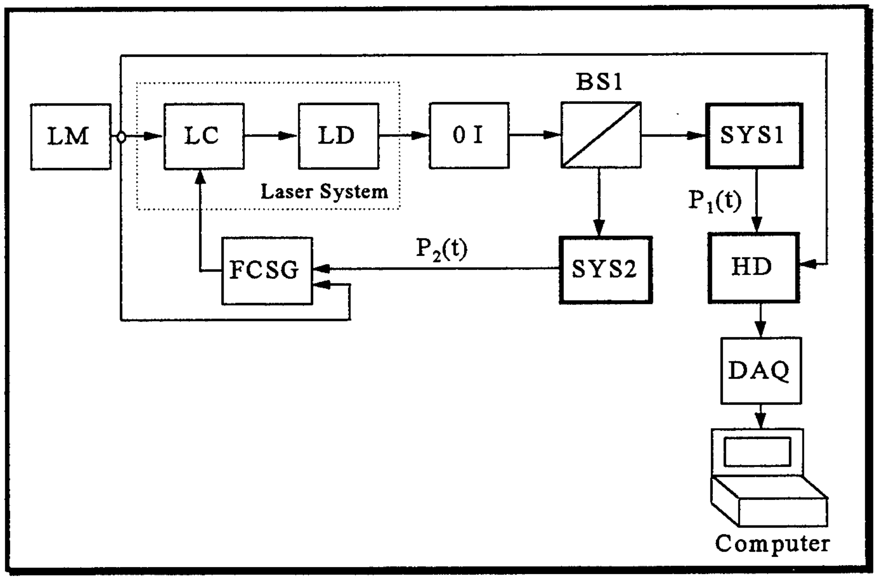


Fig. 2 Block diagram of the experimental system for SPM-LD interferometry.

As shown in Fig. 2, we configured a pair of two-beam interferometers sharing a common LD, where one (SYS1) provides an interferometric signal generated by the beam from the target, and the other (SYS2) serves as a feedback signal generator. The beam from LD was split into two beams by beam splitter BS1, and introduced into second independent Michelson interferometers. The reference interferometer has two fixed arms with a known optical path difference 2d$_r$, while the object interferometer has a fixed arm and a variable arm with a movable mirror. Let us explain how the feedback signal is generated in the reference interferometer.

The interference signal $P_2(t)$ in the reference interferometer

$$P_2(t) = P_{02}\cos\left[Z_r\cos(\omega_m t) + \theta_r + \delta_r(t)\right] \tag{15}$$

and

$$Z_r = \frac{4\pi a\beta}{\lambda_0}d_r \tag{16}$$

$$\theta_r = \frac{4\pi}{\lambda_0}d_r \tag{17}$$

$$\delta_r(t) = \frac{4\pi}{\lambda_0}(\Delta d_r) - \frac{4\pi}{\lambda_0^2}(d_r)(\Delta\lambda_{LD}) \tag{18}$$

The feedback signal $i_f(t)$ is produced by the harmonic demodulation produce in SYS2. As shown in Fig. 2, $Y_f(t)$ is fed to the amplifier, whose gain in K_p, and, we obtain the feedback signal:

$$J_1(Z_r)\sin(\theta_r + \delta_r(t)) \tag{19}$$

This feedback signal is available in the region where Z_r is between 0.5 and 3.5. When the θ_r is close to a multiple of π rad, we can stabilize the phase $\delta_r(t)$ at zero with a proportional feedback control using the feedback signal given by Eq. (19). This feedback signal $i_f(t) = A_f \sin\delta_r(t)$ is fed to injection current of the LD. Consequently, the phase change introduced by the external varied disturbance is compensated by the change in wavelength $\Delta\lambda_f = \beta i_f(t)$.

With the feedback added to the laser diode injection current, then the term $\delta_r(t)$ in Eq. (15) is actively minimized to $\Delta\delta$ as follows:

$$\delta_r(t) = \frac{4\pi}{\lambda_0}\Delta d_r - \frac{4\pi}{\lambda_0^2} d_r\left(\Delta\lambda_{LD} + \Delta\lambda_f\right) = \Delta\delta \tag{20}$$

Using the same method, we obtain the phase fluctuation $\delta_1(t)$ in SYS1 as follow:

$$\delta_1(t) = \frac{4\pi}{\lambda_0}\Delta d_1 - \frac{4\pi}{\lambda_0^2} d_1\left(\Delta\lambda_{LD} + \Delta\lambda_f\right) \tag{21}$$

By substituting Eq. (20) into Eq. (21), $\delta_1(t)$ can be expressed as [4]

$$\delta_1(t) = \frac{4\pi}{\lambda_0}\left(\Delta d_1 - \frac{d_1}{d_r}\Delta d_r\right) + \frac{d_1}{d_r}\Delta\delta \tag{22}$$

Note the same light source LD is used for both SYS1 and SYS2, and assuming that both SYS1 and SYS2 are subject to the same disturbances, then, the relationship

$$\frac{\Delta d_1}{d_1} = \frac{\Delta d_r}{d_r} \tag{23}$$

will be satisfied, thus, we get the phase $\delta_1(t)$:

$$\delta_1(t) = \frac{d_1}{d_r}\Delta\delta \tag{24}$$

Since the optical path difference $2d_1$ is not equal the optical path difference $2d_r$, the phase fluctuation $\delta_1(t)$ can be reduced less than $\Delta\delta$ if selecting $d_1 < d_r$.

4. Design of Real-time Data Processing by Computer

Our ultimate goal is to extract in real-time distance and hence the phase θ as precisely as possible, over as wide range as possible. To obtain θ in real-time, the required harmonic signals from the analog harmonic demodulator are measured by a data acquisition board in a computer.

The signals Y_1 and Y_2 are converted to 12-bit digital value by a National Instruments model AT-MIO-16E-2 data acquisition board. The AT-MIO-16E-2 board is a high-performance multifunction analog, digital, and timing I/O board for the IBM-PC computers. The sampling clock for A/D converter is synchronized with the modulation current $i_m(t)$ by AT-MIO-16E-2 board using zero-cross voltage or a selected fixed voltage values, so the sampling frequency is equal to the modulation frequency $\frac{\omega_m}{2\pi}$. In our design the actual input range and the least significant bit (LSB) of this acquisition board are selected to be, $\pm 2.5V_{pp}$ and 1.22 mV, respectively. The minimum detectable phase (MDP) can be determined primarily by one coded-phase angle $\frac{2\pi}{2^N} rad$. Since the phase is modulo 2π, we must increment or decrement the phase by 2π each time the phase crosses the positive X axis. A LabVIEW program was used to do this. Once data is acquired, a LabVIEW program processes the data according to the theory developed in Section 2 and the displacement is displayed by the computer. In an actual closed loop system the feedback logic would control the cutting tool motion.

5. The Displacement Actuators

The mirror M_2 was mounted onto a voltage controlled actuator. Because no single device could give us high resolution and wide dynamic range we have used two distinct actuators. First for high resolution displacement measurements the mirror M_2 was driven by PZT (Burleigh PZ-80) over a 6μm range. It should be noted the response of PZT to a linear ramp exhibited 5% non-linearity due to its inherent hysteresis[6]. So that, after sampling, A/D conversion and calculating displacement, the polynomial fit function is used to give the mean square error (mse). This system was also used for measurements of the noise of electrical circuit for a fixed measuring point, and to the characterize the performance of the feedback control system.

Second for wide dynamic range measurements, the object mirror M_2 was positioned by a Picomotor (New Focus Inc.) micrometer-replacement actuator that is specified at <0.1nm resolution with remote control[7]. In our system the Picomotor was operated under computer control and moved a total displacement of more than 1mm, in steps not less than 10 nm. At each step, the output voltages of the phase demodulators were sampled and converted to displacement using a LabVIEW Program that we wrote for these measurements. Then displacements are shown on computer.

6. Experimental Results

The experimental setup is shown in Fig. 2. The system contains two unbalanced Michelson interferometers: One (SYS1) is used for displacement measurement and the other (SYS2) is used for the feedback control to reduce the effects of external disturbances. The optical path difference between the two arms in SYS2 is $2d_r = 60$mm, but, in SYS1 $2d_1$ are set to different values for the two different systems. The PZT measurements were performed with $2d_1 = 60$mm and the Picomotor measurements were performed with $2d_1 \cong 110$mm, that is, the initial value of z in SYS1 is 0.6 or 1.1. A GaAIAs laser diode (SHARP LT024MD0) was used. Its' maximum output power is about 5 mW and the central wavelength λ_0 is the 785mm. The modulation efficiency β of the LD was $6.0 \times 10^{-3} nm/_{mA}$. The frequency of the phase modulation $\frac{\omega_m}{2\pi}$ was set to 10 kHz. The cutoff frequency of the LPF is set to 1 kHz to essentially eliminate the modulation frequency and it's harmonics.

As stated above, the final conversion of the output voltages from the demodulator to an actual displacement measurement and data analysis are done in software by computer. The object mirror M_2 is mounted on PZT (model PZ-80, Burleigh) or translation stage moved by a Picomotor driver (New Focus, model 8801).

Now we show some experimental results using the setup shown in Fig. 2.

First, we checked the influence of the electronic noise on the measurement system by running a series of tests with no input signal to the harmonic demodulator. Fig. 3 shows a plot of the electronic noise influence over a long period of time. As can be

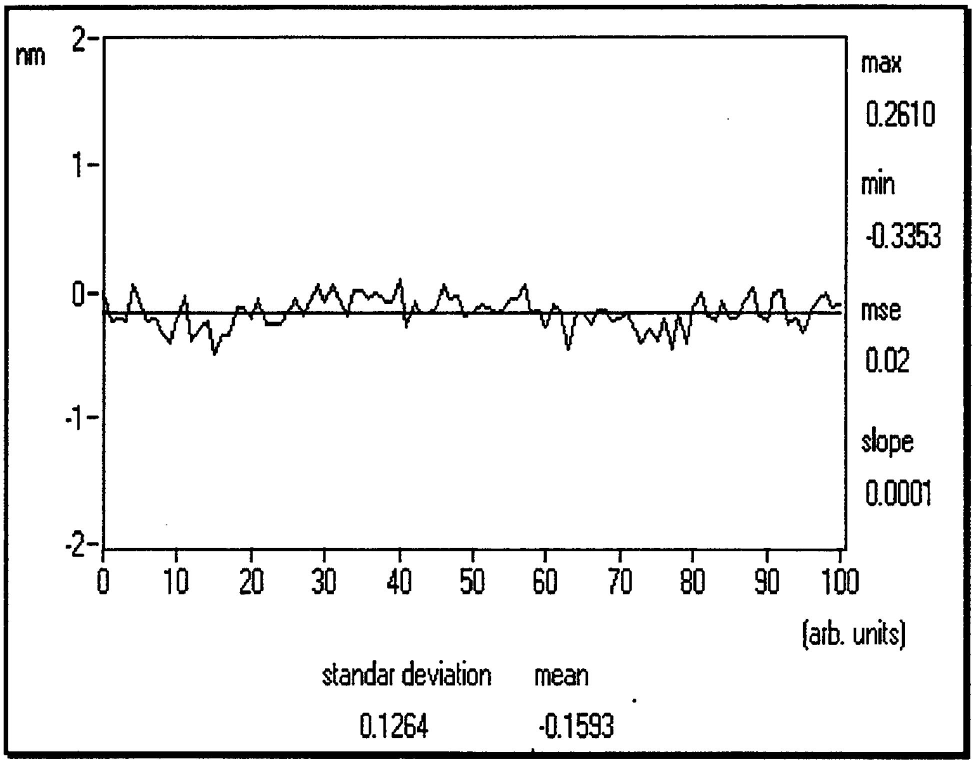


Figure 3. Measured displacement error due to electronic noise.

seen in Fig. 3. the electronics noise contributes a measurement error standard deviation of 0.13 nm. But, the minimum detectable phase (MDP) in our system can be determined primarily by one coded-phase angle $\frac{2\pi}{2^{12}}$ rad. Comparing this to the maximum spatial resolution should be $\frac{1}{2^{12}}\cdot\left(\frac{\lambda}{2}\right)$ = 0.1 nm, it is clear that the electronics noise only adds a small uncertainty to the measurement.

Next, measurements at a single point were done with and without feedback, with and without acoustic noise, with and without air flow, etc. Figs. 4 and 5 show two of these experiments. Comparing with Figs. 4 and Fig. 5, we can see that the dc drift in

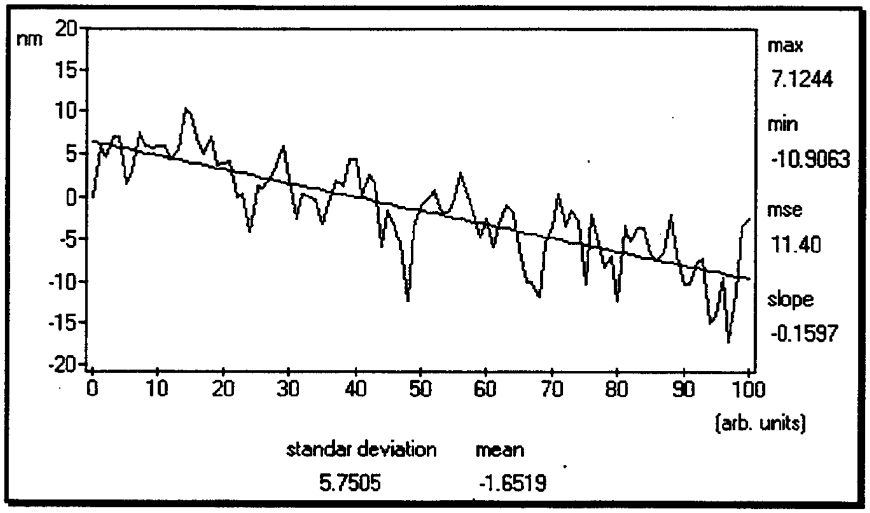


Figure 4. Single Point Measurement with Acoustic Noise and no Feedback

Fig. 4 is almost completely suppressed when feedback control was introduced, and, the standard deviation was reduced to ~ 1.3nm.

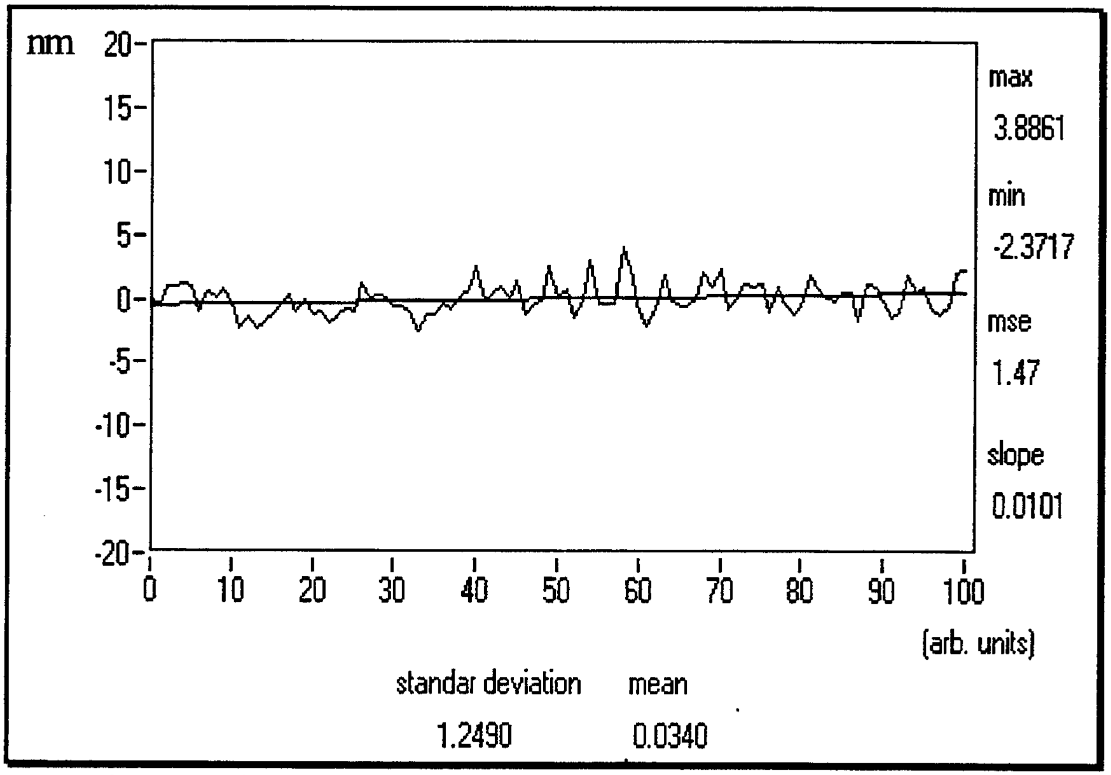
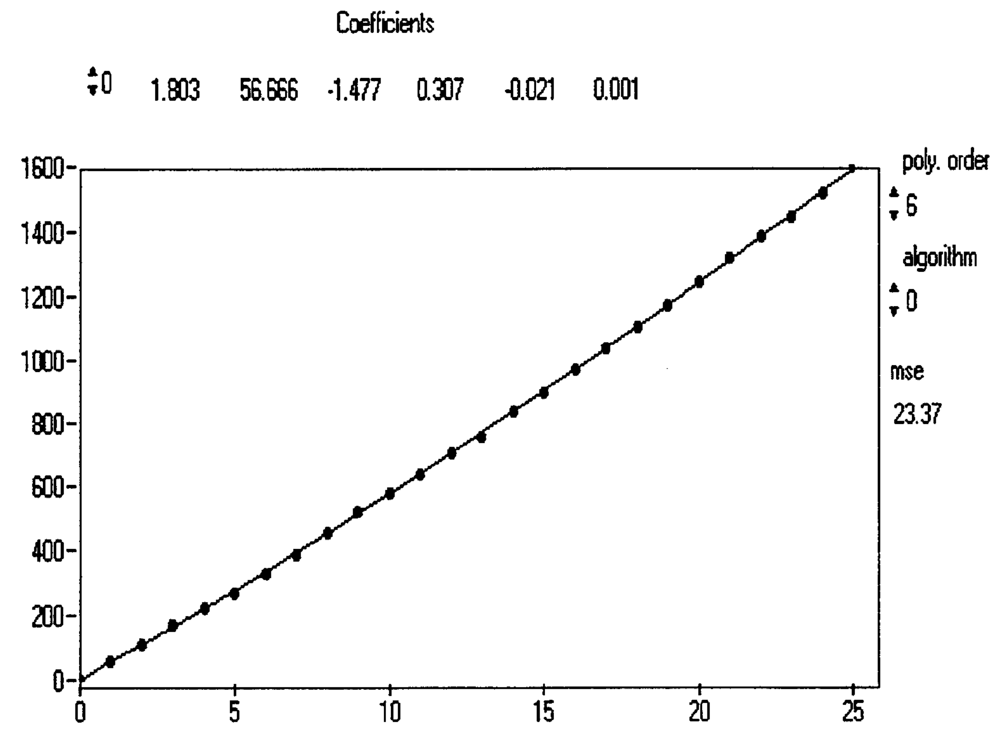


Figure 5. Single Point Measurement with Acoustic Noise and Feedback.

Third, Fig. 6 shows that a large displacement, more than 4 times $\lambda/2$, can be measured with high accuracy. In fact, it follows the nonlinear characteristic of the PZT with an accuracy of better than 5nm as shown in Fig. 6, where we used the polynomial fit function from the program to correct for the non-linearity of the PZT. The accuracy of our system was determined by taking repeated sets of data and computing the square root of the mean square error (mse) given by the polynomial fit function from the program.



Coefficients

0 1.803 56.666 -1.477 0.307 -0.021 0.001

poly. order 6

algorithm 0

mse 23.37

Figure 6. PZT movement over 250 volts.

Finally, we used a Picomotor instead of PZT to investigate the characteristic of the experiment system at larger range. With this system we demonstrated high accuracy over 0.2 mm of dynamic range.

In these experiments, we delayed the measurement by one second after each step in order to allow the mirror hardware to damp out vibrations after each step. The results of a linear fit function with the slope and mean square error calculated. The total displacement was 19130 nm or 0.191 mm, the standard deviation of each point was 44.7nm. This large standard deviation is a result of the fact that the Picomotor periodically does not move at all, when a signal is applied. This performance is within the manufacturers specifications. The measurements also, clearly shows the periodic non-linearity's in the motion of the Picomotor when the screw is rotated by 2π.

For calibration of the wide dynamic range measurements a HEIDENHAIN (Control Technics Cop.) was used also used to measure the distance traveled and the results compared to our WDR-SPM systems calculations, the results agreed within the accuracy of the HEIDENHAIN gauge within gauge's 0.1 um resolution. Long term temperature drifts caused the HEIDENHAIN gauge reading to vary by ± 0.4 μm, for large distances.

We also measured the limits of the dynamic range of our experimental system and found it to be 1mm with the hardware that we have. However, the tail, which consists of the largest distance measurements point clearly deviates from a linear fit. This is due to the limitations of the picomotor, and can easily be extended using a precision stepper motor system.

7. Discussion

7.1 Permitted Step Displacement Using Picomotor Actuator

Since the Picomotor actuator is a mechanical positioner and the mechanical stepping causes some vibration, we delayed the beginning of the A/D board sampling by 1 second so that the mechanical vibrations were damped out. This delay time can be easily changed in LabVIEW program. Once the delay time has been selected a 0.4 volt pulse is sent to the Picomotor this corresponded to a minimum step displacement of about 15 nm and an average displacement of about 20 nm. In our program, we used a 20 nm step size for our linear fit.

7.2 Limitations of dynamic range

As mentioned previously, the phase demodulator was designed to take two signal outputs which were proportional to the instantaneous sine and cosine of the phase angle of the difference optical path between two arms in SYS 1. When mirror M_2 was moved, the amplitudes of sine or cosine scaled proportionally to J_1 (z) or J_2 (z), respectively. This method may be used for J_1 (z) and J_2 (z) as long as, z, is between 0.5 and 3.5. Assuming that a = 0.1 ma, $\beta = 6 \times 10^{-3}$ nm/mA and $\lambda_0 = 785$ nm, it is possible to obtain a dynamic range of a few tens of centimeters.

Because the final conversion of the output voltages from the demodulators to an actual displacement is done in software by computer, and there always is a smallest detectable change in voltage on a data acquisition board (the change in voltage represents a least significant bit (LSB) of the digital value). For example, at our system, the LSB is required a 1.22 mV. If either output signal (either sin or cosine) from the demodulator is reduced less than 1.22 mV, no change will be detected. We also observed that the signal of the first harmonic component , falls off by a factor of 3 relative to it's initial value, then the signal was then below the sensitivity of the A/D , thus the software didn't correctly calculate displacement.

Therefore, the maximum range is determined by the photo-current on a photodetector, gain of demodulator, and LSB on a data acquisition board. In this case we were limited by the demodulator and if we increased the gain of the demodulator the maximum dynamic range will be increased for our experimental system.

8. Conclusion

We developed the first wide dynamic range SPM-LD interferometer that uses a slowly-varying approximation and built a real-time displacement measurement system with a local resolution of $\frac{2\pi}{2^{12}}$ under normal laboratory conditions. Based on this development of theory, it is not necessary to keep the amplitudes of the first, the second harmonic component equal in experiment, so it is easy to implement displacement measurement with 1mm range. Due to use feedback control system to reduce external disturbances, this method is insensitive to the wavelength fluctuation of LD, the gain drift, and the temperature change or vibration of external circumstance. The standard deviation of 5nm for PZT, the dynamic range of 1 mm for Picomotor actuator has been demonstrated with the same experimental system. The use of the LabVIEW software for data acquisition and instrument control, made it relatively easy to implement a real-time displacement measurement. Our calibration for Picomotor is more directly and accuracy than Ref. [7]. If LPF's gain can be improved then the dynamic range can be increased to 10.

This is the first wide dynamic range high resolution distance measuring system that can tolerate vibrations of several hundred nm. The system is suitable for real time closed loop operation in a high precision machining environment.

References

1. J.M. Wagner and J.B. Spicer, "Theoretical noise-limited sensitivity of classical interferometry ", J. Opt. Soc. Am. B, 4, 1316 (1987)

2. A.J. den Boef, "Two-Wavelength scanning spot interferometer using single-frequency diode lasers," Appl. Opt. 27, 306-311 (1988)

3. A. Chebbour, C. Gorecki, G. Tribillon, "Range finding and velocimetry with directional discrimination using a modulated laser diode Mechelson interferometer," Opt. Commun. 111, 1-5 (1994)

4. Osami Sasaki, Kazuchide Takahashi, Takamasa Suzuki, "Sinusoidal phase modulating laser diode interferometer with a feedback control system to eliminate external disturbance," Optical Engineering, 29, 1511-1515 (1990).

5. Takamasa Suzuki, Osami Sasaki, S. Takayama, T. Maruyama, "Real-time displacement measurement using synchronous detection in a sinusoidal phase modulating interferometer," Optical Engineering, 32, 1033-1037 (1993).

6. Burleigh Instruments, Inc. "Instruction Manual," p. 9.

7. New Focus, Inc. "Application Note 4, Picomotor drivers: A guide to computer control and closed-loop applications."

TASK 3.5

PUBLICATIONS AND TECHNICAL REPORTS PUBLISHED

TASK 3.5 PUBLICATIONS AND TECHNICAL REPORTS PUBLISHED

There were no publications, nor any technical reports published, as a result of Task 3.5 research, during the life of this grant. A Masters Degree Thesis was published, but is unavailable for copying and submittal in concert with the timing of this report. It will be submitted separately and expeditiously. A conference paper has been accepted for CLEO '97. It is in preparation and will be submitted upon presentation.

TASK 3.5

PARTICIPATING SCIENTIFIC PERSONNEL

TASK 3.5 PARTICIPATING SCIENTIFIC PERSONNEL

NAME	STATUS
Lawrence S. Alvarez	Engineer II
Chi K. Man	MS Awarded
Dr. T. M. Shay	Professor
Dr. Y. Wu	Post-Doctoral Fellow
Dr. B. Yin	Post-Doctoral Fellow

REPORT OF INVENTIONS

There are no inventions to be reported under Grant No. DAAHO4-93-G-0508, as of this final report submittal. There is a reasonable possibility that a patent will be applied for, as a consequence of the Task 3.5 research. In the event that this occurs, the ARO will be notified formally and expeditiously.